Chapter 7

The Unsymmetric Eigenvalue Problem

Having discussed linear equations and least squares, we now direct our attention to the third major problem area in matrix computations, the algebraic eigenvalue problem. The unsymmetric problem is considered in this chapter and the more agreeable symmetric case in the next.

Our first task is to present the decompositions of Schur and Jordan along with the basic properties of eigenvalues and invariant subspaces. The contrasting behavior of these two decompositions sets the stage for §7.2 in which we investigate how the eigenvalues and invariant subspaces of a matrix are affected by perturbation. Condition numbers are developed that permit estimation of the errors that can be expected to arise because of roundoff.

The key algorithm of the chapter is the justly famous QR algorithm. This procedure is the most complex algorithm presented in this book and its development is spread over three sections. We derive the basic QR iteration in §7.3 as a natural generalization of the simple power method. The next

two sections are devoted to making this basic iteration computationally feasible. This involves the introduction of the Hessenberg decomposition in §7.4 and the notion of origin shifts in §7.5.

The QR algorithm computes the real Schur form of a matrix, a canonical form that displays eigenvalues but not eigenvectors. Consequently, additional computations usually must be performed if information regarding invariant subspaces is desired. In §7.6, which could be subtitled, "What to Do after the Real Schur Form is Calculated," we discuss various invariant subspace calculations that can follow the QR algorithm.

Finally, in the last section we consider the generalized eigenvalue problem $Ax = \lambda Bx$ and a variant of the QR algorithm that has been devised to solve it. This algorithm, called the QZ algorithm, underscores the importance of orthogonal matrices in the eigenproblem, a central theme of the chapter.

It is appropriate at this time to make a remark about complex versus real arithmetic. In this book, we focus on the development of real arithmetic algorithms for real matrix problems. This chapter is no exception even though a real unsymmetric matrix can have complex eigenvalues. However, in the derivation of the practical, real arithmetic QR algorithm and in the mathematical analysis of the eigenproblem itself, it is convenient to work in the complex field. Thus, the reader will find that we have switched to complex notation in §7.1, §7.2, and §7.3. In these sections, we use complex versions of the QR factorization, the singular value decomposition, and the CS decomposition.

Before You Begin

Chapters 1-3 and §§5.1-5.2 are assumed. Within this chapter there are the following dependencies:

§7.1 → §7.2 → §7.3 → §7.4 → §7.5 → §7.6 → §7.7

Complementary references include Fox (1964), Wilkinson (1965), Gourlay and Watson (1973), Stewart (1973), Hager (1988), Ciarlet (1989), Stewart and Sun (1990), Watkins (1991), Saad (1992), Jennings and Mc Keowen (1992), Datta (1995), Trefethen and Bau (1997), and Demmel (1996). Some Matlab functions important to this chapter are `eig`, `poly`, `polyeig`, `hess`, `qz`, `rsf2csf`, `cdf2rdf`, `schur`, and `balance`. LAPACK connections include

LAPACK: Unsymmetric Eigenproblem	
_GEBAL	Balance transform
_GEBAK	Undo balance transform
_GEHRD	Hessenberg reduction $U^H AV = H$
_ORMHR	U (factored form) times matrix (real case)
_ORGHR	Generates U (real case)
_UNMHR	U (factored form) times matrix (complex case)
_UNGHR	Generates U (complex case)
_HSEQR	Schur decomposition of Hessenberg matrix
_HSEIN	Eigenvectors of Hessenberg matrix by inverse iteration
_GEES	Schur decomp of general matrix with e.value ordering
_GEESX	Same but with condition estimates
_GEEV	Eigenvalues and left and right eigenvectors of general matrix
_GEEVX	Same but with condition estimates
_TREVC	Selected eigenvectors of upper quasitriangular matrix
_TRSNA	Cond. estimates of selected eigenvalues of upper quasitriangular matrix
_TREXC	Unitary reordering of Schur decomposition
_TRSEN	Same but with condition estimates
_TRSYL	Solves $AX + XB = C$ for upper quasitriangular A and B

LAPACK: Unsymmetric Generalized Eigenproblem	
_GGBAL	Balance transform
_GGHRD	Reduction to Hessenberg-Triangular form
_HGEQZ	Generalized Schur decomposition
_TGEVC	Eigenvectors
_GGBAK	Undo balance transform

7.1 Properties and Decompositions

In this section we survey the mathematical background necessary to develop and analyze the eigenvalue algorithms that follow.

7.1.1 Eigenvalues and Invariant Subspaces

The *eigenvalues* of a matrix $A \in \mathbb{C}^{n\times n}$ are the n roots of its *characteristic polynomial* $p(z) = \det(zI - A)$. The set of these roots is called the *spectrum* and is denoted by $\lambda(A)$. If $\lambda(A) = \{\lambda_1, \ldots, \lambda_n\}$, then it follows that

$$\det(A) = \lambda_1\lambda_2\cdots\lambda_n\,.$$

Moreover, if we define the *trace* of A by

$$\mathrm{tr}(A) \;=\; \sum_{i=1}^{n} a_{ii},$$

then $\mathrm{tr}(A) = \lambda_1 + \cdots + \lambda_n$. This follows by looking at the coefficient of z^{n-1} in the characteristic polynomial.

If $\lambda \in \lambda(A)$, then the nonzero vectors $x \in \mathbb{C}^n$ that satisfy

$$Ax = \lambda x$$

are referred to as *eigenvectors*. More precisely, x is a *right eigenvector* for λ if $Ax = \lambda x$ and a *left eigenvector* if $x^H A = \lambda x^H$. Unless otherwise stated, "eigenvector" means "right eigenvector."

An eigenvector defines a one-dimensional subspace that is invariant with respect to premultiplication by A. More generally, a subspace $S \subseteq \mathbb{C}^n$ with the property that

$$x \in S \Longrightarrow Ax \in S$$

is said to be *invariant* (for A). Note that if

$$AX = XB, \qquad B \in \mathbb{C}^{k\times k},\ X \in \mathbb{C}^{n\times k},$$

then $\text{ran}(X)$ is invariant and $By = \lambda y \Rightarrow A(Xy) = \lambda(Xy)$. Thus, if X has full column rank, then $AX = XB$ implies that $\lambda(B) \subseteq \lambda(A)$. If X is square and nonsingular, then $\lambda(A) = \lambda(B)$ and we say that A and $B = X^{-1}AX$ are *similar*. In this context, X is called a *similarity transformation*.

7.1.2 Decoupling

Many eigenvalue computations involve breaking the given problem down into a collection of smaller eigenproblems. The following result is the basis for these reductions.

Lemma 7.1.1 *If $T \in \mathbb{C}^{n\times n}$ is partitioned as follows,*

$$T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$$
$$\qquad\qquad p \quad\ q$$

then $\lambda(T) = \lambda(T_{11}) \cup \lambda(T_{22})$.

Proof. Suppose

$$Tx = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where $x_1 \in \mathbb{C}^p$ and $x_2 \in \mathbb{C}^q$. If $x_2 \neq 0$, then $T_{22}x_2 = \lambda x_2$ and so $\lambda \in \lambda(T_{22})$. If $x_2 = 0$, then $T_{11}x_1 = \lambda x_1$ and so $\lambda \in \lambda(T_{11})$. It follows that $\lambda(T) \subset \lambda(T_{11}) \cup \lambda(T_{22})$. But since both $\lambda(T)$ and $\lambda(T_{11}) \cup \lambda(T_{22})$ have the same cardinality, the two sets are equal. □

7.1.3 The Basic Unitary Decompositions

By using similarity transformations, it is possible to reduce a given matrix to any one of several canonical forms. The canonical forms differ in how they display the eigenvalues and in the kind of invariant subspace information that they provide. Because of their numerical stability we begin by discussing the reductions that can be achieved with unitary similarity.

Lemma 7.1.2 *If $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{p\times p}$, and $X \in \mathbb{C}^{n\times p}$ satisfy*

$$AX = XB, \qquad \text{rank}(X) = p, \tag{7.1.1}$$

then there exists a unitary $Q \in \mathbb{C}^{n\times n}$ such that

$$Q^H AQ = T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix} \tag{7.1.2}$$
$$\qquad\qquad\quad p \quad n-p$$

where $\lambda(T_{11}) = \lambda(A) \cap \lambda(B)$.

Proof. Let

$$X = Q\begin{bmatrix} R_1 \\ 0 \end{bmatrix} \qquad Q \in \mathbb{C}^{n\times n},\ R_1 \in \mathbb{C}^{p\times p}$$

be a QR factorization of X. By substituting this into (7.1.1) and rearranging we have

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} B$$

where

$$Q^H AQ = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix} .$$
$$\qquad p \quad n-p$$

By using the nonsingularity of R_1 and the equations $T_{21}R_1 = 0$ and $T_{11}R_1 = R_1B$, we can conclude that $T_{21} = 0$ and $\lambda(T_{11}) = \lambda(B)$. The conclusion now follows because from Lemma 7.1.1 $\lambda(A) = \lambda(T) = \lambda(T_{11}) \cup \lambda(T_{22})$. □

Example 7.1.1 If

$$A = \begin{bmatrix} 67.00 & 177.60 & -63.20 \\ -20.40 & 95.88 & -87.16 \\ 22.80 & 67.84 & 12.12 \end{bmatrix},$$

$X = [20,\ -9,\ -12]^T$ and $B = [25]$, then $AX = XB$. Moreover, if the orthogonal matrix Q is defined by

$$Q = \begin{bmatrix} -.800 & .360 & .480 \\ .360 & .928 & -.096 \\ .480 & -.096 & .872 \end{bmatrix},$$

then $Q^TX = [-25,\ 0,\ 0]^T$ and

$$Q^TAQ = T = \begin{bmatrix} 25 & -90 & 5 \\ 0 & 147 & -104 \\ 0 & 146 & 3 \end{bmatrix}.$$

A calculation shows that $\lambda(A) = \{25, 75 + 100i, 75 - 100i\}$.

Lemma 7.1.2 says that a matrix can be reduced to block triangular form using unitary similarity transformations if we know one of its invariant subspaces. By induction we can readily establish the decomposition of Schur (1909).

Theorem 7.1.3 (Schur Decomposition) *If $A \in \mathbb{C}^{n\times n}$, then there exists a unitary $Q \in \mathbb{C}^{n\times n}$ such that*

$$Q^H AQ = T = D + N \tag{7.1.3}$$

where $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $N \in \mathbb{C}^{n\times n}$ is strictly upper triangular. Furthermore, Q can be chosen so that the eigenvalues λ_i appear in any order along the diagonal.

Proof. The theorem obviously holds when $n = 1$. Suppose it holds for all matrices of order $n - 1$ or less. If $Ax = \lambda x$, where $x \neq 0$, then by Lemma 7.1.2 (with $B = (\lambda)$) there exists a unitary U such that:

$$U^H AU = \begin{bmatrix} \lambda & w^H \\ 0 & C \end{bmatrix} \begin{matrix} 1 \\ n-1 \end{matrix} \; .$$
$$\qquad\qquad 1 \quad n-1$$

By induction there is a unitary $\tilde{U}$ such that $\tilde{U}^H C \tilde{U}$ is upper triangular. Thus, if $Q = U\text{diag}(1, \tilde{U})$, then $Q^H AQ$ is upper triangular. $\square$

Example 7.1.2 If

$$A = \begin{bmatrix} 3 & 8 \\ -2 & 3 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} .8944i & .4472 \\ -.4472 & -.8944i \end{bmatrix},$$

then Q is unitary and

$$Q^H AQ = \begin{bmatrix} 3+4i & -6 \\ 0 & 3-4i \end{bmatrix}.$$

If $Q = [\, q_1, \ldots, q_n \,]$ is a column partitioning of the unitary matrix Q in (7.1.3), then the q_i are referred to as *Schur vectors*. By equating columns in the equations $AQ = QT$ we see that the Schur vectors satisfy

$$Aq_k = \lambda_k q_k + \sum_{i=1}^{k-1} n_{ik} q_i \qquad k = 1{:}n\,. \tag{7.1.4}$$

From this we conclude that the subspaces

$$S_k = \text{span}\{q_1, \ldots, q_k\} \qquad k = 1{:}n$$

are invariant. Moreover, it is not hard to show that if $Q_k = [\, q_1, \ldots, q_k \,]$, then $\lambda(Q_k^H AQ_k) = \{\lambda_1, \ldots, \lambda_k\}$. Since the eigenvalues in (7.1.3) can be arbitrarily ordered, it follows that there is at least one k-dimensional invariant subspace associated with each subset of k eigenvalues.

Another conclusion to be drawn from (7.1.4) is that the Schur vector q_k is an eigenvector if and only if the k-th column of N is zero. This turns out to be the case for $k = 1{:}n$ whenever $A^H A = AA^H$. Matrices that satisfy this property are called *normal*.

Corollary 7.1.4 *$A \in \mathbb{C}^{n\times n}$ is normal if and only if there exists a unitary $Q \in \mathbb{C}^{n\times n}$ such that $Q^H AQ = \text{diag}(\lambda_1, \ldots, \lambda_n)$.*

Proof. It is easy to show that if A is unitarily similar to a diagonal matrix, then A is normal. On the other hand, if A is normal and $Q^H AQ = T$ is its Schur decomposition, then T is also normal. The corollary follows by showing that a normal, upper triangular matrix is diagonal. □

Note that if $Q^H AQ = T = \text{diag}(\lambda_i) + N$ is a Schur decomposition of a general n-by-n matrix A, then $\| N \|_F$ is independent of the choice of Q:

$$\| N \|_F^2 = \| A \|_F^2 - \sum_{i=1}^{n} |\lambda_i|^2 \equiv \Delta^2(A).$$

This quantity is referred to as A's *departure from normality.* Thus, to make T "more diagonal," it is necessary to rely on nonunitary similarity transformations.

7.1.4 Nonunitary Reductions

To see what is involved in nonunitary similarity reduction, we examine the block diagonalization of a 2-by-2 block triangular matrix.

Lemma 7.1.5 *Let $T \in \mathbb{C}^{n\times n}$ be partitioned as follows:*

$$T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{matrix} p \\ q \end{matrix} \quad .$$
$$\qquad\quad p \quad\; q$$

Define the linear transformation $\phi: \mathbb{C}^{p\times q} \to \mathbb{C}^{p\times q}$ by

$$\phi(X) = T_{11}X - XT_{22}$$

where $X \in \mathbb{C}^{p\times q}$. Then ϕ is nonsingular if and only if $\lambda(T_{11}) \cap \lambda(T_{22}) = \emptyset$. If ϕ is nonsingular and Y is defined by

$$Y = \begin{bmatrix} I_p & Z \\ 0 & I_q \end{bmatrix} \qquad \phi(Z) = -T_{12}$$

then $Y^{-1}TY = \text{diag}(T_{11}, T_{22})$.

Proof. Suppose $\phi(X) = 0$ for $X \neq 0$ and that

$$U^H XV = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{matrix} r \\ p-r \end{matrix}$$
$$\qquad\quad r \quad q-r$$

is the SVD of X with $\Sigma_r = \text{diag}(\sigma_i)$, $r = \text{rank}(X)$. Substituting this into the equation $T_{11}X = XT_{22}$ gives

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

where $U^H T_{11} U = (A_{ij})$ and $V^H T_{22} V = (B_{ij})$. By comparing blocks we see that $A_{21} = 0$, $B_{12} = 0$, and $\lambda(A_{11}) = \lambda(B_{11})$. Consequently,

$$\emptyset \neq \lambda(A_{11}) = \lambda(B_{11}) \subseteq \lambda(T_{11}) \cap \lambda(T_{22}).$$

On the other hand, if $\lambda \in \lambda(T_{11}) \cap \lambda(T_{22})$ then we have nonzero vectors x and y so $T_{11}x = \lambda x$ and $y^H T_{22} = \lambda y^H$. A calculation shows that $\phi(xy^H) = 0$. Finally, if ϕ is nonsingular then the matrix Z above exists and

$$\begin{aligned} Y^{-1}TY &= \begin{bmatrix} I & -Z \\ 0 & I \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} T_{11} & T_{11}Z - ZT_{22} + T_{12} \\ 0 & T_{22} \end{bmatrix} = \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix}. \square \end{aligned}$$

Example 7.1.3 If

$$T = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 3 & 8 \\ 0 & -2 & 3 \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} 1.0 & 0.5 & -0.5 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

then

$$Y^{-1}TY = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 8 \\ 0 & -2 & 3 \end{bmatrix}.$$

By repeatedly applying Lemma 7.1.5, we can establish the following more general result:

Theorem 7.1.6 (Block Diagonal Decomposition) *Suppose*

$$Q^H A Q = T = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1q} \\ 0 & T_{22} & \cdots & T_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{qq} \end{bmatrix} \tag{7.1.5}$$

is a Schur decomposition of $A \in \mathbb{C}^{n\times n}$ and assume that the T_{ii} are square. If $\lambda(T_{ii}) \cap \lambda(T_{jj}) = \emptyset$ whenever $i \neq j$, then there exists a nonsingular matrix $Y \in \mathbb{C}^{n\times n}$ such that

$$(QY)^{-1}A(QY) = \text{diag}(T_{11}, \ldots, T_{qq}). \tag{7.1.6}$$

Proof. A proof can be obtained by using Lemma 7.1.5 and induction. □

If each diagonal block T_{ii} is associated with a distinct eigenvalue, then we obtain

Corollary 7.1.7 *If $A \in \mathbb{C}^{n\times n}$ then there exists a nonsingular X such that*

$$X^{-1}AX = \text{diag}(\lambda_1 I + N_1, \ldots, \lambda_q I + N_q) \qquad N_i \in \mathbb{C}^{n_i\times n_i} \qquad (7.1.7)$$

where $\lambda_1, \ldots, \lambda_q$ are distinct, the integers $n_1, \ldots, n_q$ satisfy $n_1 + \cdots + n_q = n$, and each N_i is strictly upper triangular.

A number of important terms are connected with decomposition (7.1.7). The integer n_i is referred to as the *algebraic multiplicity* of λ_i. If $n_i = 1$, then λ_i is said to be *simple* . The *geometric multiplicity* of λ_i equals the dimensions of null(N_i), i.e., the number of linearly independent eigenvectors associated with λ_i. If the algebraic multiplicity of λ_i exceeds its geometric multiplicity, then λ_i is said to be a *defective eigenvalue.* A matrix with a defective eigenvalue is referred to as a *defective matrix.* Nondefective matrices are also said to be *diagonalizable* in light of the following result:

Corollary 7.1.8 (Diagonal Form) *$A \in \mathbb{C}^{n\times n}$ is nondefective if and only if there exists a nonsingular $X \in \mathbb{C}^{n\times n}$ such that*

$$X^{-1}AX = \text{diag}(\lambda_1, \ldots, \lambda_n). \qquad (7.1.8)$$

Proof. A is nondefective if and only if there exist independent vectors $x_1 \ldots x_n \in \mathbb{C}^n$ and scalars $\lambda_1, \ldots, \lambda_n$ such that $Ax_i = \lambda_i x_i$ for $i = 1{:}n$. This is equivalent to the existence of a nonsingular $X = [\, x_1, \ldots, x_n \,] \in \mathbb{C}^{n\times n}$ such that $AX = XD$ where $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$. □

Note that if y_i^H is the ith row of X^{-1}, then $y_i^H A = \lambda_i y_i^H$. Thus, the columns of X^{-T} are left eigenvectors and the columns of X are right eigenvectors.

Example 7.1.4 If

$$A = \begin{bmatrix} 5 & -1 \\ -2 & 6 \end{bmatrix} \quad \text{and} \quad X = \begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix}$$

then $X^{-1}AX = \text{diag}(4, 7)$.

If we partition the matrix X in (7.1.7),

$$X = \begin{bmatrix} \underset{n_1}{X_1}, & \ldots, & \underset{n_q}{X_q} \end{bmatrix}$$

then $\mathbb{C}^n = \text{ran}(X_1) \oplus \ldots \oplus \text{ran}(X_q)$, a direct sum of invariant subspaces. If the bases for these subspaces are chosen in a special way, then it is possible to introduce even more zeroes into the upper triangular portion of $X^{-1}AX$.

Theorem 7.1.9 (Jordan Decomposition) *If $A \in \mathbb{C}^{n\times n}$, then there exists a nonsingular $X \in \mathbb{C}^{n\times n}$ such that $X^{-1}AX = \text{diag}(J_1, \ldots, J_t)$ where*

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \cdots & 0 \\ 0 & \lambda_i & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \cdots & & 0 & \lambda_i \end{bmatrix}$$

is m_i-by-m_i and $m_1 + \cdots + m_t = n$.

Proof. See Halmos (1958, pp. 112 ff.) □

The J_i are referred to as *Jordan blocks* . The number and dimensions of the Jordan blocks associated with each distinct eigenvalue is unique, although their ordering along the diagonal is not.

7.1.5 Some Comments on Nonunitary Similarity

The Jordan block structure of a defective matrix is difficult to determine numerically. The set of n-by-n diagonalizable matrices is dense in $\mathbb{C}^{n\times n}$, and thus, small changes in a defective matrix can radically alter its Jordan form. We have more to say about this in §7.6.5.

A related difficulty that arises in the eigenvalue problem is that a nearly defective matrix can have a poorly conditioned matrix of eigenvectors. For example, any matrix X that diagonalizes

$$A = \begin{bmatrix} 1+\epsilon & 1 \\ 0 & 1-\epsilon \end{bmatrix} \qquad 0 < \epsilon \ll 1 \qquad (7.1.9)$$

has a 2-norm condition of order $1/\epsilon$.

These observations serve to highlight the difficulties associated with ill-conditioned similarity transformations. Since

$$fl(X^{-1}AX) = X^{-1}AX + E, \qquad (7.1.10)$$

where

$$\| E \|_2 \approx \mathbf{u}\kappa_2(X)\| A \|_2 \qquad (7.1.11)$$

is it clear that large errors can be introduced into an eigenvalue calculation when we depart from unitary similarity.

7.1.6 Singular Values and Eigenvalues

Since the singular values of A and its Schur decomposition $Q^H AQ = \text{diag}(\lambda_i) + N$ are the same, it follows that

$$\sigma_{min}(A) \leq \min_i |\lambda_i| \leq \max_i |\lambda_i| \leq \sigma_{max}(A).$$

From what we know about the condition of triangular matrices, it may be the case that

$$\max_{i,j} \frac{|\lambda_i|}{|\lambda_j|} \ll \kappa_2(A).$$

This is a reminder that for nonnormal matrices, eigenvalues do not have the "predictive power" of singular values when it comes to $Ax = b$ sensitivity matters. Eigenvalues of nonnormal matrices have other shortcomings. See §11.3.4.

Problems

P7.1.1 Show that if $T \in \mathbb{C}^{n \times n}$ is upper triangular and normal, then T is diagonal.

P7.1.2 Verify that if X diagonalizes the 2-by-2 matrix in (7.1.9) and $\epsilon \leq 1/2$ then $\kappa_1(X) \geq 1/\epsilon$.

P7.1.3 Suppose $A \in \mathbb{C}^{n \times n}$ has distinct eigenvalues. Show that if $Q^H AQ = T$ is its Schur decomposition and $AB = BA$, then $Q^H BQ$ is upper triangular.

P7.1.4 Show that if A and B^H are in $\mathbb{C}^{m \times n}$ with $m \geq n$, then:

$$\lambda(AB) = \lambda(BA) \cup \{ \underbrace{0, \ldots, 0}_{m-n} \}.$$

P7.1.5 Given $A \in \mathbb{C}^{n \times n}$, use the Schur decomposition to show that for every $\epsilon > 0$, there exists a diagonalizable matrix B such that $\| A - B \|_2 \leq \epsilon$. This shows that the set of diagonalizable matrices is dense in $\mathbb{C}^{n \times n}$ and that the Jordan canonical form is not a continuous matrix decomposition.

P7.1.6 Suppose $A_k \to A$ and that $Q_k^H A_k Q_k = T_k$ is a Schur decomposition of A_k. Show that $\{Q_k\}$ has a converging subsequence $\{Q_{k_i}\}$ with the property that

$$\lim_{i \to \infty} Q_{k_i} = Q$$

where $Q^H AQ = T$ is upper triangular. This shows that the eigenvalues of a matrix are continuous functions of its entries.

P7.1.7 Justify (7.1.10) and (7.1.11).

P7.1.8 . Show how to compute the eigenvalues of

$$M = \begin{bmatrix} A & C \\ B & D \end{bmatrix} \begin{matrix} k \\ j \end{matrix}$$
$$\qquad\quad \begin{matrix} k & j \end{matrix}$$

where A, B, C, and D are given real diagonal matrices.

P7.1.9 Use the JCF to show that if all the eigenvalues of a matrix A are strictly less

than unity, then $\lim_{k\to\infty} A^k = 0$.

P7.1.10 The initial value problem

$$\begin{array}{rclcl} \dot{x}(t) & = & y(t) & & x(0) = 1 \\ \dot{y}(t) & = & -x(t) & & y(0) = 0 \end{array}$$

has solution $x(t) = \cos(t)$ and $y(t) = \sin(t)$. Let $h > 0$. Here are three reasonable iterations that can be used to compute approximations $x_k \approx x(kh)$ and $y_k \approx y(kh)$ assuming that $x_0 = 1$ and $y_k = 0$:

$$\text{Method 1:} \quad \begin{array}{rcl} x_{k+1} & = & 1 + hy_k \\ y_{k+1} & = & 1 - hx_k \end{array}$$

$$\text{Method 2:} \quad \begin{array}{rcl} x_{k+1} & = & 1 + hy_k \\ y_{k+1} & = & 1 - hx_{k+1} \end{array}$$

$$\text{Method 3:} \quad \begin{array}{rcl} x_{k+1} & = & 1 + hy_{k+1} \\ y_{k+1} & = & 1 - hx_{k+1} \end{array}$$

Express each method in the form

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = A_h \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

where A_h is a 2-by-2 matrix. For each case, compute $\lambda(A_h)$ and use the previous problem to discuss $\lim x_k$ and $\lim y_k$ as $k \to \infty$.

P7.1.11 If $J \in \mathbb{R}^{d\times d}$ is a Jordan block, what is $\kappa_\infty(J)$?

P7.1.12 Show that if

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$$
$$\qquad\qquad p \quad\; q$$

is normal and $\lambda(R_{11}) \cap \lambda(R_{22}) = \emptyset$, then $R_{12} = 0$.

Notes and References for Sec. 7.1

The mathematical properties of the algebraic eigenvalue problem are elegantly covered in Wilkinson (1965, chapter 1) and Stewart (1973, chapter 6). For those who need further review we also recommend

R. Bellman (1970). *Introduction to Matrix Analysis,* 2nd ed., McGraw-Hill, New York.

I.C. Gohberg, P. Lancaster, and L. Rodman (1986). *Invariant Subspaces of Matrices With Applications,* John Wiley and Sons, New York.

M. Marcus and H. Minc (1964). *A Survey of Matrix Theory and Matrix Inequalities,* Allyn and Bacon, Boston.

L. Mirsky (1963). *An Introduction to Linear Algebra* , Oxford University Press, Oxford.

The Schur decomposition originally appeared in

I. Schur (1909). "On the Characteristic Roots of a Linear Substitution with an Application to the Theory of Integral Equations." *Math. Ann. 66,* 488-510 (German).

A proof very similar to ours is given on page 105 of

H.W. Turnbull and A.C. Aitken (1961). *An Introduction to the Theory of Canonical Forms,* Dover, New York.

Connections between singular values, eigenvalues, and pseudoeigenvalues (see §11.3.4) are discussed in

K-C. Toh and L.N. Trefethen (1994). "Pseudozeros of Polynomials and Pseudospectra of Companion Matrices," *Numer. Math. 68*, 403–425.
F. Kittaneh (1995). "Singular Values of Companion Matrices and Bounds on Zeros of Polynomials," *SIAM J. Matrix Anal. Appl. 16*, 333–340.

7.2 Perturbation Theory

The act of computing eigenvalues is the act of computing zeros of the characteristic polynomial. Galois theory tells us that such a process has to be iterative if $n > 4$ and so errors will arise because of finite termination. In order to develop intelligent stopping criteria we need an informative perturbation theory that tells us how to think about approximate eigenvalues and invariant subspaces.

7.2.1 Eigenvalue Sensitivity

Several eigenvalue routines produce a sequence of similarity transformations X_k with the property that the matrices $X_k^{-1}AX_k$ are progressively "more diagonal." The question naturally arises, how well do the diagonal elements of a matrix approximate its eigenvalues?

Theorem 7.2.1 (Gershgorin Circle Theorem) *If* $X^{-1}AX = D + F$ *where* $D = \text{diag}(d_1, \ldots, d_n)$ *and* F *has zero diagonal entries, then*

$$\lambda(A) \subseteq \bigcup_{i=1}^{n} D_i$$

where $D_i = \{z \in \mathbb{C} : |z - d_i| \leq \sum_{j=1}^{n} |f_{ij}|\}$.

Proof. Suppose $\lambda \in \lambda(A)$ and assume without loss of generality that $\lambda \neq d_i$ for $i = 1{:}n$. Since $(D - \lambda I) + F$ is singular, it follows from Lemma 2.3.3 that

$$1 \leq \| (D - \lambda I)^{-1} F \|_\infty = \sum_{j=1}^{n} \frac{|f_{kj}|}{|d_k - \lambda|}$$

for some k, $1 \leq k \leq n$. But this implies that $\lambda \in D_k$. □

It can also be shown that if the Gershgorin disk D_i is isolated from the other disks, then it contains precisely one of A's eigenvalues. See Wilkinson (1965,

pp.71ff.).

Example 7.2.1 If

$$A = \begin{bmatrix} 10 & 2 & 3 \\ -1 & 0 & 2 \\ 1 & -2 & 1 \end{bmatrix}$$

then $\lambda(A) \approx \{10.226,\ .3870 + 2.2216i,\ .3870 - 2.2216i\}$ and the Gershgorin disks are $D_1 = \{\ |z| : |z - 10| \leq 5\}$, $D_2 = \{\ |z| : |z| \leq 3\}$, and $D_3 = \{\ |z| : |z - 1| \leq 3\}$.

For some very important eigenvalue routines it is possible to show that the computed eigenvalues are the exact eigenvalues of a matrix $A + E$ where E is small in norm. Consequently, we must understand how the eigenvalues of a matrix can be affected by small perturbations. A sample result that sheds light on this issue is the following theorem.

Theorem 7.2.2 (Bauer-Fike) *If μ is an eigenvalue of $A + E \in \mathbb{C}^{n\times n}$ and $X^{-1}AX = D = \text{diag}(\lambda_1, \ldots, \lambda_n)$, then*

$$\min_{\lambda\in\lambda(A)} |\lambda - \mu| \leq \kappa_p(X)\| E \|_p$$

where $\| \cdot \|_p$ denotes any of the p-norms.

Proof. We need only consider the case when μ is not in $\lambda(A)$. If the matrix $X^{-1}(A + E - \mu I)X$ is singular, then so is $I + (D - \mu I)^{-1}(X^{-1}EX)$. Thus, from Lemma 2.3.3 we obtain

$$1 \leq \| (D - \mu I)^{-1}(X^{-1}EX) \|_p \leq \| (D - \mu I)^{-1} \|_p \| X \|_p \| E \|_p \| X^{-1} \|_p .$$

Since $(D - \mu I)^{-1}$ is diagonal and the p-norm of a diagonal matrix is the absolute value of the largest diagonal entry, it follows that

$$\| (D - \mu I)^{-1} \|_p = \min_{\lambda\in\lambda(A)} \frac{1}{|\lambda - \mu|}$$

from which the theorem follows. □

An analogous result can be obtained via the Schur decomposition:

Theorem 7.2.3 *Let $Q^H AQ = D+N$ be a Schur decomposition of $A \in \mathbb{C}^{n\times n}$ as in (7.1.3). If $\mu \in \lambda(A + E)$ and p is the smallest positive integer such that $|N|^p = 0$, then*

$$\min_{\lambda\in\lambda(A)} |\lambda - \mu| \leq \max(\theta,\ \theta^{1/p})$$

where

$$\theta = \| E \|_2 \sum_{k=0}^{p-1} \| N \|_2^k .$$

Proof. Define

$$\delta = \min_{\lambda\in\lambda(A)} |\lambda - \mu| = \frac{1}{\| (\mu I - D)^{-1} \|_2}.$$

The theorem is clearly true if $\delta = 0$. If $\delta > 0$ then $I - (\mu I - A)^{-1}E$ is singular and by Lemma 2.3.3 we have

$$\begin{aligned} 1 &\leq \| (\mu I - A)^{-1}E \|_2 \leq \| (\mu I - A)^{-1} \|_2 \| E \|_2 \qquad (7.2.1) \\ &= \| ((\mu I - D) - N)^{-1} \|_2 \| E \|_2 . \end{aligned}$$

Since $(\mu I - D)^{-1}$ is diagonal and $|N|^p = 0$ it is not hard to show that $((\mu I - D)^{-1}N)^p = 0$. Thus,

$$((\mu I - D) - N)^{-1} = \sum_{k=0}^{p-1} \left((\mu I - D)^{-1}N\right)^k (\mu I - D)^{-1}$$

and so

$$\| ((\mu I - D) - N)^{-1} \|_2 \leq \frac{1}{\delta} \sum_{k=0}^{p-1} \left(\frac{\| N \|_2}{\delta}\right)^k .$$

If $\delta > 1$ then

$$\| (\mu I - D) - N)^{-1} \|_2 \leq \frac{1}{\delta} \sum_{k=0}^{p-1} \| N \|_2^k$$

and so from (7.2.1), $\delta \leq \theta$. If $\delta \leq 1$ then

$$\| (\mu I - D) - N)^{-1} \|_2 \leq \frac{1}{\delta^p} \sum_{k=0}^{p-1} \| N \|_2^k$$

and so from (7.2.1), $\delta^p \leq \theta$. Thus, $\delta \leq \max(\theta, \theta^{1/p})$. □

Example 7.2.2 If

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 4.001 \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ .001 & 0 & 0 \end{bmatrix},$$

then $\lambda(A + E) \approx \{1.0001,\ 4.0582,\ 3.9427\}$ and A's matrix of eigenvectors satisfies $\kappa_2(X) \approx 10^7$. The Bauer-Fike bound in Theorem 7.2.2 has order 10^4, while the Schur bound in Theorem 7.2.3 has order 10^0.

Theorems 7.2.2 and 7.2.3 each indicate potential eigenvalue sensitivity if A is nonnormal. Specifically, if $\kappa_2(X)$ or $\| N \|_2^{p-1}$ is large, then small changes in A can induce large changes in the eigenvalues.

Example 7.2.3 If

$$A = \begin{bmatrix} 0 & I_9 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} 0 & 0 \\ 10^{-10} & 0 \end{bmatrix},$$

then for all $\lambda \in \lambda(A)$ and $\mu \in \lambda(A+E)$, $|\lambda - \mu| = 10^{-1}$. In this example a change of order 10^{-10} in A results in a change of order 10^{-1} in its eigenvalues.

7.2.2 The Condition of a Simple Eigenvalue

Extreme eigenvalue sensitivity for a matrix A cannot occur if A is normal. On the other hand, nonnormality does not necessarily imply eigenvalue sensitivity. Indeed, a nonnormal matrix can have a mixture of well-conditioned and ill-conditioned eigenvalues. For this reason, it is beneficial to refine our perturbation theory so that it is applicable to individual eigenvalues and not the spectrum as a whole.

To this end, suppose that λ is a simple eigenvalue of $A \in \mathbb{C}^{n\times n}$ and that x and y satisfy $Ax = \lambda x$ and $y^H A = \lambda y^H$ with $\| x \|_2 = \| y \|_2 = 1$. If $Y^H AX = J$ is the Jordan decomposition with $Y^H = X^{-1}$, then y and x are nonzero multiples of $X(:,i)$ and $Y(:,i)$ for some i. It follows from $1 = Y(:,i)^H X(:,i)$ that $y^H x \neq 0$, a fact that we shall use shortly.

Using classical results from function theory, it can be shown that in a neighborhood of the origin there exist differentiable $x(\epsilon)$ and $\lambda(\epsilon)$ such that

$$(A + \epsilon F)x(\epsilon) = \lambda(\epsilon)x(\epsilon) \qquad \| F \|_2 = 1$$

where $\lambda(0) = \lambda$ and $x(0) = x$. By differentiating this equation with respect to ϵ and setting $\epsilon = 0$ in the result, we obtain

$$A\dot{x}(0) + Fx \;=\; \dot{\lambda}(0)x + \lambda\dot{x}(0)\,.$$

Applying y^H to both sides of this equation, dividing by $y^H x$, and taking absolute values gives

$$|\dot{\lambda}(0)| \;=\; \left|\frac{y^H F x}{y^H x}\right| \;\leq\; \frac{1}{|y^H x|}.$$

The upper bound is attained if $F = yx^H$. For this reason we refer to the reciprocal of

$$s(\lambda) \;=\; |y^H x|$$

as the *condition of the eigenvalue* λ.

Roughly speaking, the above analysis shows that if order ϵ perturbations are made in A, then an eigenvalue λ may be perturbed by an amount $\epsilon/s(\lambda)$. Thus, if $s(\lambda)$ is small, then λ is appropriately regarded as ill-conditioned. Note that $s(\lambda)$ is the cosine of the angle between the left and right eigenvectors associated with λ and is unique only if λ is simple.

A small $s(\lambda)$ implies that A is near a matrix having a multiple eigenvalue. In particular, if λ is distinct and $s(\lambda) < 1$, then there exists an E such that λ is a repeated eigenvalue of $A + E$ and

$$\frac{\| E \|_2}{\| A \|_2} \;\leq\; \frac{s(\lambda)}{\sqrt{1 - s(\lambda)^2}}\,.$$

This result is proved in Wilkinson (1972).

Example 7.2.4 If

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 4.001 \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ .001 & 0 & 0 \end{bmatrix},$$

then $\lambda(A+E) \approx \{1.0001,\ 4.0582,\ 3.9427\}$ and $s(1) \approx .8 \times 10^0$, $s(4) \approx .2 \times 10^{-3}$, and $s(4.001) \approx .2 \times 10^{-3}$. Observe that $\| E \|_2 / s(\lambda)$ is a good estimate of the perturbation that each eigenvalue undergoes.

7.2.3 Sensitivity of Repeated Eigenvalues

If λ is a repeated eigenvalue, then the eigenvalue sensitivity question is more complicated. For example, if

$$A = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad F = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

then $\lambda(A + \epsilon F) = \{1 \pm \sqrt{\epsilon a}\}$. Note that if $a \neq 0$, then it follows that the eigenvalues of $A + \epsilon F$ are not differentiable at zero; their rate of change at the origin is infinite. In general, if λ is a defective eigenvalue of A, then $O(\epsilon)$ perturbations in A can result in $O(\epsilon^{1/p})$ perturbations in λ if λ is associated with a p-dimensional Jordan block. See Wilkinson (1965, pp. 77ff.) for a more detailed discussion.

7.2.4 Invariant Subspace Sensitivity

A collection of sensitive eigenvectors can define an insensitive invariant subspace provided the corresponding cluster of eigenvalues is isolated. To be precise, suppose

$$Q^H A Q = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} \qquad (7.2.2)$$
$$\qquad\qquad\quad r \quad n-r$$

is a Schur decomposition of A with

$$Q = [\, Q_1 \quad Q_2 \,] \qquad (7.2.3)$$
$$\qquad r \quad n-r$$

It is clear from our discussion of eigenvector perturbation that the sensitivity of the invariant subspace $\text{ran}(Q_1)$ depends on the distance between $\lambda(T_{11})$ and $\lambda(T_{22})$. The proper measure of this distance turns out to be the smallest singular value of the linear transformation $X \rightarrow T_{11}X - XT_{22}$.

(Recall that this transformation figures in Lemma 7.1.5.) In particular, if we define the *separation* between the matrices T_{11} and T_{22} by

$$\text{sep}(T_{11}, T_{22}) \;=\; \min_{X \neq 0} \frac{\| T_{11}X - XT_{22} \|_F}{\| X \|_F}, \qquad (7.2.4)$$

then we have the following general result:

Theorem 7.2.4 *Suppose that (7.2.2) and (7.2.3) hold and that for any matrix $E \in \mathbb{C}^{n \times n}$ we partition $Q^H EQ$ as follows:*

$$Q^H EQ \;=\; \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} \; . \qquad \begin{matrix} r & n-r \end{matrix}$$

If $\text{sep}(T_{11}, T_{22}) > 0$ *and*

$$\| E \|_2 \left(1 + \frac{5\| T_{12} \|_2}{\text{sep}(T_{11}, T_{22})}\right) \;\leq\; \frac{\text{sep}(T_{11}, T_{22})}{5},$$

then there exists a $P \in \mathbb{C}^{(n-r)\times r}$ *with*

$$\| P \|_2 \;\leq\; 4\frac{\| E_{21} \|_2}{\text{sep}(T_{11}, T_{22})}$$

such that the columns of $\hat{Q}_1 = (Q_1 + Q_2 P)(I + P^H P)^{-1/2}$ *are an orthonormal basis for a subspace invariant for* $A + E$.

Proof. This result is a slight recasting of Theorem 4.11 in Stewart (1973) which should be consulted for proof details. See also Stewart and Sun (1990, p.230). The matrix $(I + P^H P)^{-1/2}$ is the inverse of the square root of the symmetric positive definite matrix $I + P^H P$. See §4.2.10. □

Corollary 7.2.5 *If the assumptions in Theorem 7.2.4 hold, then*

$$\text{dist}(\text{ran}(Q_1), \text{ran}(\hat{Q}_1)) \;\leq\; 4\frac{\| E_{21} \|_2}{\text{sep}(T_{11}, T_{22})}.$$

Proof. Using the SVD of P, it can be shown that

$$\| P(I + P^H P)^{-1/2} \|_2 \leq \| P \|_2. \qquad (7.2.5)$$

The corollary follows because the required distance is the norm of $Q_2^H \hat{Q}_1 = P(I + P^H P)^{-1/2}$. □

Thus, the reciprocal of $\text{sep}(T_{11}, T_{22})$ can be thought of as a condition number that measures the sensitivity of $\text{ran}(Q_1)$ as an invariant subspace.

Example 7.2.5 Suppose

$$T_{11} = \begin{bmatrix} 3 & 10 \\ 0 & 1 \end{bmatrix}, \quad T_{22} = \begin{bmatrix} 0 & -20 \\ 0 & 3.01 \end{bmatrix}, \quad \text{and} \quad T_{12} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

and that

$$A = T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}.$$

Observe that $AQ_1 = Q_1T_{11}$ where $Q_1 = [e_1, e_2] \in \mathbb{R}^{4\times 2}$. A calculation shows that $\text{sep}(T_{11}, T_{22}) \approx .0003$. If

$$E_{21} = 10^{-6}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

and we examine the Schur decomposition of

$$A + E = \begin{bmatrix} T_{11} & T_{12} \\ E_{21} & T_{22} \end{bmatrix},$$

then we find that Q_1 gets perturbed to

$$\hat{Q}_1 = \begin{bmatrix} -.9999 & -.0003 \\ .0003 & -.9999 \\ -.0005 & -.0026 \\ .0000 & .0003 \end{bmatrix}.$$

Thus, we have $\text{dist}(\text{ran}(\hat{Q}_1), \text{ran}(Q_1)) \approx .0027 \approx 10^{-6}/\text{sep}(T_{11}, T_{22})$.

7.2.5 Eigenvector Sensitivity

If we set $r = 1$ in the preceding subsection, then the analysis addresses the issue of eigenvector sensitivity.

Corollary 7.2.6 *Suppose $A, E \in \mathbb{C}^{n\times n}$ and that $Q = [\, q_1\, Q_2\,] \in \mathbb{C}^{n\times n}$ is unitary with $q_1 \in \mathbb{C}^n$. Assume*

$$Q^HAQ = \begin{bmatrix} \lambda & v^H \\ 0 & T_{22} \end{bmatrix} \begin{matrix} 1 \\ n-1 \end{matrix} \qquad Q^HEQ = \begin{bmatrix} \epsilon & \gamma^H \\ \delta & E_{22} \end{bmatrix} \begin{matrix} 1 \\ n-1 \end{matrix}.$$
$$\qquad\quad 1 \quad n-1 \qquad\qquad\qquad\qquad 1 \quad n-1$$

(Thus, q_1 is an eigenvector.) If $\sigma = \sigma_{min}(T_{22} - \lambda I) > 0$ and

$$\| E \|_2 \left(1 + \frac{5\| v \|_2}{\sigma}\right) \le \frac{\sigma}{5},$$

then there exists $p \in \mathbb{C}^{n-1}$ with

$$\| p \|_2 \le 4\frac{\| \delta \|_2}{\sigma}$$

such that $\hat{q}_1 = (q_1 + Q_2p)/\sqrt{1 + p^Hp}$ is a unit 2-norm eigenvector for $A+E$. Moreover,

$$\text{dist}(\text{span}\{q_1\}, \text{span}\{\hat{q}_1\}) \le 4\frac{\| \delta \|_2}{\sigma}.$$

Proof. The result follows from Theorem 7.2.4, Corollary 7.2.5 and the observation that if $T_{11} = \lambda$, then $\text{sep}(T_{11}, T_{22}) = \sigma_{min}(T_{22} - \lambda I)$. □

Note that $\sigma_{min}(T_{22} - \lambda I)$ roughly measures the separation of λ from the eigenvalues of T_{22}. We have to say "roughly" because

$$\text{sep}(\lambda, T_{22}) = \sigma_{min}(T_{22} - \lambda I) \leq \min_{\mu \in \lambda(T_{22})} |\mu - \lambda|$$

and the upper bound can be a gross overestimate.

That the separation of the eigenvalues should have a bearing upon eigenvector sensitivity should come as no surprise. Indeed, if λ is a nondefective, repeated eigenvalue, then there are an infinite number of possible eigenvector bases for the associated invariant subspace. The preceding analysis merely indicates that this indeterminancy begins to be felt as the eigenvalues coalesce. In other words, the eigenvectors associated with nearby eigenvalues are "wobbly."

Example 7.2.6 If

$$A = \begin{bmatrix} 1.01 & 0.01 \\ 0.00 & 0.99 \end{bmatrix}$$

then the eigenvalue $\lambda = .99$ has condition $1/s(.99) \approx 1.118$ and associated eigenvector $x = [.4472, \ -.8944]^T$. On the other hand, the eigenvalue $\hat{\lambda} = 1.00$ of the "nearby" matrix

$$A + E = \begin{bmatrix} 1.01 & 0.01 \\ 0.00 & 1.00 \end{bmatrix}$$

has an eigenvector $\hat{x} = [.7071, \ -.7071]^T$.

Problems

P7.2.1 Suppose $Q^H AQ = \text{diag}(\lambda_1) + N$ is a Schur decomposition of $A \in \mathbb{C}^{n\times n}$ and define $\nu(A) = \| A^H A - AA^H \|_F$. The upper and lower bounds in

$$\frac{\nu(A)^2}{6\| A \|_F^2} \leq \| N \|_F^2 \leq \sqrt{\frac{n^3 - n}{12}}\nu(A)$$

are established by Henrici (1962) and Eberlein (1965), respectively. Verify these results for the case $n = 2$.

P7.2.2 Suppose $A \in \mathbb{C}^{n\times n}$ and $X^{-1}AX = \text{diag}(\lambda_1, \ldots, \lambda_n)$ with distinct λ_i. Show that if the columns of X have unit 2-norm, then $\kappa_F(X)^2 = n\sum_{i=1}^n (1/s(\lambda_i))^2$

P7.2.3 Suppose $Q^H AQ = \text{diag}(\lambda_i) + N$ is a Schur decomposition of A and that $X^{-1}AX = \text{diag}\ (\lambda_i)$. Show $\kappa_2(X)^2 \geq 1 + (\| N \|_F / \| A \|_F)^2$. See Loizou (1969).

P7.2.4 If $X^{-1}AX = \text{diag}\ (\lambda_i)$ and $|\lambda_1| \geq \cdots \geq |\lambda_n|$, then

$$\frac{\sigma_i(A)}{\kappa_2(X)} \leq |\lambda_i| \leq \kappa_2(X)\sigma_i(A)\,.$$

Prove this result for the $n = 2$ case. See Ruhe (1975).

P7.2.5 Show that if $A = \begin{bmatrix} a & c \\ 0 & b \end{bmatrix}$ and $a \neq b$, then $s(a) = s(b) = (1 + |c/(a-b)|^2)^{-1/2}$.

P7.2.6 Suppose

$$A = \begin{bmatrix} \lambda & v^T \\ 0 & T_{22} \end{bmatrix}$$

and that $\lambda \notin \lambda(T_{22})$. Show that if $\sigma = \text{sep}(\lambda, T_{22})$, then

$$s(\lambda) = \frac{1}{\sqrt{1 + \| (T_{22} - \lambda I)^{-1} v \|_2^2}} \le \frac{\sigma}{\sqrt{\sigma^2 + \| v \|_2^2}}.$$

P7.2.7 Show that the condition of a simple eigenvalue is preserved under unitary similarity transformations.

P7.2.8 With the same hypothesis as in the Bauer-Fike theorem (Theorem 7.2.2), show that $\min_{\lambda \in \lambda(A)} |\lambda - \mu| \le \| \, |X^{-1}| \, |E| \, |X| \, \|_p$.

P7.2.9 Verify (7.2.5).

P7.2.10 Show that if $B \in \mathbb{C}^{m \times m}$ and $C \in \mathbb{C}^{n \times n}$, then sep$(B, C)$ is less than or equal to $|\lambda - \mu|$ for all $\lambda \in \lambda(B)$ and $\mu \in \lambda(C)$.

Notes and References for Sec. 7.2

Many of the results presented in this section may be found in Wilkinson (1965, chapter 2), Stewart and Sun (1990) as well as in

F.L. Bauer and C.T. Fike (1960). "Norms and Exclusion Theorems," *Numer. Math. 2*, 123–44.

A.S. Householder (1964). *The Theory of Matrices in Numerical Analysis*. Blaisdell, New York.

The following papers are concerned with the effect of perturbations on the eigenvalues of a general matrix:

A. Ruhe (1970). "Perturbation Bounds for Means of Eigenvalues and Invariant Subspaces," *BIT 10*, 343–54.

A. Ruhe (1970). "Properties of a Matrix with a Very Ill-Conditioned Eigenproblem," *Numer. Math. 15*, 57–60.

J.H. Wilkinson (1972). "Note on Matrices with a Very Ill-Conditioned Eigenproblem," *Numer. Math. 19*, 176 78.

W. Kahan, B.N. Parlett, and E. Jiang (1982). "Residual Bounds on Approximate Eigensystems of Nonnormal Matrices," *SIAM J. Numer. Anal. 19*, 470–484.

J.H. Wilkinson (1984). "On Neighboring Matrices with Quadratic Elementary Divisors," *Numer. Math. 44*, 1-21.

J.V. Burke and M.L. Overton (1992). "Stable Perturbations of Nonsymmetric Matrices," *Lin.Alg. and Its Application 171*, 249–273.

Wilkinson's work on nearest defective matrices is typical of a growing body of literature that is concerned with "nearness" problems. See

N.J. Higham (1985). "Nearness Problems in Numerical Linear Algebra," PhD Thesis, University of Manchester, England.

C. Van Loan (1985). "How Near is a Stable Matrix to an Unstable Matrix?," *Contemporary Mathematics, Vol. 47*, 465–477.

J.W. Demmel (1987). "On the Distance to the Nearest Ill-Posed Problem," *Numer. Math. 51*, 251–289.

J.W. Demmel (1987). "A Counterexample for two Conjectures About Stability," *IEEE Trans. Auto. Cont.* *AC-32*, 340–342.

A. Ruhe (1987). "Closest Normal Matrix Found!," *BIT 27*, 585-598.

R. Byers (1988). "A Bisection Method for Measuring the Distance of a Stable Matrix to the Unstable Matrices," *SIAM J. Sci. and Stat. Comp. 9*, 875 881.

J.W. Demmel (1988). "The Probability that a Numerical Analysis Problem is Difficult," *Math. Comp. 50*, 449–480.

N.J. Higham (1989). "Matrix Nearness Problems and Applications," in *Applications of Matrix Theory*, M.J.C. Gover and S. Barnett (eds), Oxford University Press, Oxford UK, 1 27.

Aspects of eigenvalue condition are discussed in

C. Van Loan (1987). "On Estimating the Condition of Eigenvalues and Eigenvectors," *Lin. Alg. and Its Applic.* 88/89, 715–732.

C.D. Meyer and G.W. Stewart (1988). "Derivatives and Perturbations of Eigenvectors," *SIAM J. Num. Anal. 25*, 679–691.

G.W. Stewart and G. Zhang (1991). "Eigenvalues of Graded Matrices and the Condition Numbers of Multiple Eigenvalues," *Numer. Math. 58*, 703–712.

J.-G. Sun (1992). "On Condition Numbers of a Nondefective Multiple Eigenvalue," *Numer. Math. 61*, 265–276.

The relationship between the eigenvalue condition number, the departure from normality, and the condition of the eigenvector matrix is discussed in

P. Henrici (1962). "Bounds for Iterates, Inverses, Spectral Variation and Fields of Values of Non-normal Matrices," *Numer. Math. 4*, 24–40.

P. Eberlein (1965). "On Measures of Non Normality for Matrices," *Amer. Math. Soc. Monthly 72*, 995 96.

R.A. Smith (1967). "The Condition Numbers of the Matrix Eigenvalue Problem," *Numer. Math. 10* 232 40.

G. Loizou (1969). "Nonnormality and Jordan Condition Numbers of Matrices," *J. ACM 16*, 580 40.

A. van der Sluis (1975). "Perturbations of Eigenvalues of Non normal Matrices," *Comm. ACM 18*, 30 36.

The paper by Henrici also contains a result similar to Theorem 7.2.3. Penetrating treatments of invariant subspace perturbation include

T. Kato (1966). *Perturbation Theory for Linear Operators*, Springer-Verlag, New York.

C. Davis and W.M. Kahan (1970). "The Rotation of Eigenvectors by a Perturbation, III," *SIAM J. Num. Anal. 7*, 1 46.

G.W. Stewart (1971). "Error Bounds for Approximate Invariant Subspaces of Closed Linear Operators," *SIAM. J. Num. Anal. 8*, 796 808.

G.W. Stewart (1973). "Error and Perturbation Bounds for Subspaces Associated with Certain Eigenvalue Problems," *SIAM Review 15*, 727 64.

Detailed analyses of the function sep(.,.) and the map $X \rightarrow AX + XA^T$ are given in

J. Varah (1979). "On the Separation of Two Matrices," *SIAM J. Num. Anal. 16*, 216–22.

R. Byers and S.G. Nash (1987). "On the Singular Vectors of the Lyapunov Operator," *SIAM J. Alg. and Disc. Methods 8*, 59–66.

Gershgorin's Theorem can be used to derive a comprehensive perturbation theory. See Wilkinson (1965, chapter 2). The theorem itself can be generalized and extended in various ways; see

R.S. Varga (1970). "Minimal Gershgorin Sets for Partitioned Matrices," *SIAM J. Num. Anal.* 7, 493–507.
R.J. Johnston (1971). "Gershgorin Theorems for Partitioned Matrices," *Lin. Alg. and Its Applic.* 4, 205–20.

7.3 Power Iterations

Suppose that we are given $A \in \mathbb{C}^{n\times n}$ and a unitary $U_0 \in \mathbb{C}^{n\times n}$. Assume that Householder orthogonalization (Algorithm 5.2.1) can be extended to complex matrices (it can) and consider the following iteration:

$$
\begin{aligned}
&T_0 = U_0^H A U_0 \\
&\textbf{for } k = 1, 2, \ldots \\
&\qquad T_{k-1} = U_k R_k \quad \text{(QR factorization)} \\
&\qquad T_k = R_k U_k \\
&\textbf{end}
\end{aligned}
\tag{7.3.1}
$$

Since $T_k = R_k U_k = U_k^H (U_k R_k) U_k = U_k^H T_{k-1} U_k$ it follows by induction that

$$
T_k = (U_0 U_1 \cdots U_k)^H A (U_0 U_1 \cdots U_k). \tag{7.3.2}
$$

Thus, each T_k is unitarily similar to A. Not so obvious, and what is the central theme of this section, is that the T_k almost always converge to upper triangular form. That is, (7.3.2) almost always "converges" to a Schur decomposition of A.

Iteration (7.3.1) is called the *QR iteration*, and it forms the backbone of the most effective algorithm for computing the Schur decomposition. In order to motivate the method and to derive its convergence properties, two other eigenvalue iterations that are important in their own right are presented first: the power method and the method of orthogonal iteration.

7.3.1 The Power Method

Suppose $A \in \mathbb{C}^{n\times n}$ is diagonalizable, that $X^{-1}AX = \text{diag}(\lambda_1, \ldots, \lambda_n)$ with $X = [\, x_1, \ldots, x_n \,]$, and $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Given a unit 2-norm $q^{(0)} \in \mathbb{C}^n$, the *power method* produces a sequence of vectors $q^{(k)}$ as follows:

$$
\begin{aligned}
&\textbf{for } k = 1, 2, \ldots \\
&\qquad z^{(k)} = A q^{(k-1)} \\
&\qquad q^{(k)} = z^{(k)} / \| z^{(k)} \|_2 \\
&\qquad \lambda^{(k)} = [q^{(k)}]^H A q^{(k)} \\
&\textbf{end}
\end{aligned}
\tag{7.3.3}
$$

There is nothing special about doing a 2-norm normalization except that it imparts a greater unity on the overall discussion in this section.

Let us examine the convergence properties of the power iteration. If

$$q^{(0)} = a_1x_1 + a_2x_2 + \cdots + a_nx_n$$

and $a_1 \neq 0$, then it follows that

$$A^kq^{(0)} = a_1\lambda_1^k \left(x_1 + \sum_{j=2}^{n} \frac{a_j}{a_1} \left(\frac{\lambda_j}{\lambda_1}\right)^k x_j \right).$$

Since $q^{(k)} \in \text{span}\{A^kq^{(0)}\}$ we conclude that

$$\text{dist}\left(\text{span}\{q^{(k)}\}, \text{span}\{x_1\}\right) = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

and moreover,

$$|\lambda_1 - \lambda^{(k)}| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right).$$

If $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$ then we say that λ_1 is a *dominant eigenvalue*. Thus, the power method converges if λ_1 is dominant and if $q^{(0)}$ has a component in the direction of the corresponding *dominant eigenvector* x_1.

The behavior of the iteration without these assumptions is discussed in Wilkinson (1965, p.570) and Parlett and Poole (1973).

Example 7.3.1 If

$$A = \begin{bmatrix} -261 & 209 & -49 \\ -530 & 422 & -98 \\ -800 & 631 & -144 \end{bmatrix}$$

then $\lambda(A) = \{10, 4, 3\}$. Applying (7.3.3) with $q^{(0)} = [1, 0, 0]^T$ we find

k	$\lambda^{(k)}$
1	13.0606
2	10.7191
3	10.2073
4	10.0633
5	10.0198
6	10.0063
7	10.0020
8	10.0007
9	10.0002

In practice, the usefulness of the power method depends upon the ratio $|\lambda_2|/|\lambda_1|$, since it dictates the rate of convergence. The danger that $q^{(0)}$ is deficient in x_1 is a less worrisome matter because rounding errors sustained during the iteration typically ensure that the subsequent $q^{(k)}$ have a component in this direction. Moreover, it is typically the case in applications

where the dominant eigenvalue and eigenvector are desired that an à priori estimate of x_1 is known. Normally, by setting $q^{(0)}$ to be this estimate, the dangers of a small a_1 are minimized.

Note that the only thing required to implement the power method is a subroutine capable of computing matrix-vector products of the form Aq. It is not necessary to store A in an n-by-n array. For this reason, the algorithm can be of interest when A is large and sparse and when there is a sufficient gap between $|\lambda_1|$ and $|\lambda_2|$.

Estimates for the error $|\lambda^{(k)} - \lambda_1|$ can be obtained by applying the perturbation theory developed in the previous section. Define the vector $r^{(k)} = Aq^{(k)} - \lambda^{(k)}q^{(k)}$ and observe that $(A + E^{(k)})q^{(k)} = \lambda^{(k)}q^{(k)}$ where $E^{(k)} = -r^{(k)}[q^{(k)}]^H$. Thus $\lambda^{(k)}$ is an eigenvalue of $A + E^{(k)}$ and

$$| \lambda^{(k)} - \lambda_1 | \approx \frac{\| E^{(k)} \|_2}{s(\lambda_1)} = \frac{\| r^{(k)} \|_2}{s(\lambda_1)} .$$

If we use the power method to generate approximate right *and* left dominant eigenvectors, then it is possible to obtain an estimate of $s(\lambda_1)$. In particular, if $w^{(k)}$ is a unit 2-norm vector in the direction of $(A^H)^k w^{(0)}$, then we can use the approximation $s(\lambda_1) \approx | {w^{(k)}}^H q^{(k)} |$.

7.3.2 Orthogonal Iteration

A straightforward generalization of the power method can be used to compute higher-dimensional invariant subspaces. Let r be a chosen integer satisfying $1 \le r \le n$. Given an n-by-r matrix Q_0 with orthonormal columns, the method of *orthogonal iteration* generates a sequence of matrices $\{Q_k\} \subseteq \mathbb{C}^{n\times r}$ as follows:

$$\begin{aligned} &\textbf{for } k = 1, 2, \ldots \\ &\qquad Z_k = AQ_{k-1} \\ &\qquad Q_k R_k = Z_k \qquad \text{(QR factorization)} \\ &\textbf{end} \end{aligned} \tag{7.3.4}$$

Note that if $r = 1$, then this is just the power method. Moreover, the sequence $\{Q_k e_1\}$ is precisely the sequence of vectors produced by the power iteration with starting vector $q^{(0)} = Q_0 e_1$.

In order to analyze the behavior of this iteration, suppose that

$$Q^H A Q = T = \text{diag}(\lambda_i) + N \qquad |\lambda_1| \ge |\lambda_2| \ge \cdots \ge |\lambda_n| \tag{7.3.5}$$

is a Schur decomposition of $A \in \mathbb{C}^{n\times n}$. Assume that $1 \le r < n$ and partition Q, T, and N as follows:

$$Q = [\, \underset{r}{Q_\alpha} \quad \underset{n-r}{Q_\beta} \,] \qquad\qquad T = \begin{array}{cc} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} & \begin{matrix} r \\ n-r \end{matrix} \\ \begin{matrix} r & n-r \end{matrix} & \end{array} \tag{7.3.6}$$

$$N = \begin{bmatrix} N_{11} & N_{12} \\ 0 & N_{22} \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} \quad .$$
$$\begin{matrix} r & n-r \end{matrix}$$

If $|\lambda_r| > |\lambda_{r+1}|$, then the subspace $D_r(A) = \text{ran}(Q_\alpha)$ is said to be a *dominant* invariant subspace. It is the unique invariant subspace associated with the eigenvalues $\lambda_1, \ldots, \lambda_r$. The following theorem shows that with reasonable assumptions, the subspaces $\text{ran}(Q_k)$ generated by (7.3.4) converge to $D_r(A)$ at a rate proportional to $|\lambda_{r+1}/\lambda_r|^k$.

Theorem 7.3.1 *Let the Schur decomposition of $A \in \mathbb{C}^{n\times n}$ be given by (7.3.5) and (7.3.6) with $n \geq 2$. Assume that $|\lambda_r| > |\lambda_{r+1}|$ and that $\theta \geq 0$ satisfies*

$$(1+\theta)|\lambda_r| \;>\; \| N \|_F \,.$$

If $Q_0 \in \mathbb{C}^{n\times r}$ has orthonormal columns and

$$d \;=\; \text{dist}(D_r(A^H), \text{ran}(Q_0)) \;<\; 1,$$

then the matrices Q_k generated by (7.3.4) satisfy

$$\text{dist}(D_r(A), \text{ran}(Q_k)) \;\leq$$
$$\frac{(1+\theta)^{n-2}}{\sqrt{1-d^2}}\left(1 + \frac{\| T_{12} \|_F}{\text{sep}(T_{11}, T_{22})}\right)\left(\frac{|\lambda_{r+1}| + \| N \|_F/(1+\theta)}{|\lambda_r| - \| N \|_F/(1+\theta)}\right)^k .$$

Proof. The proof is given in an appendix at the end of this section. □

The condition $d < 1$ in Theorem 7.3.1 ensures that the initial Q matrix is not deficient in certain eigendirections:

$$d < 1 \;\leftrightarrow\; D_r(A^H)^\perp \cap \text{ran}(Q_0) = \{0\}.$$

The theorem essentially says that if this condition holds and if θ is chosen large enough, then

$$\text{dist}(D_r(A), \text{ran}(Q_k)) \;\leq\; c\left|\frac{\lambda_{r+1}}{\lambda_r}\right|^k$$

where c depends on $\text{sep}(T_{11}, T_{22})$ and A's departure from normality. Needless to say, convergence can be very slow if the gap between $|\lambda_r|$ and $|\lambda_{r+1}|$ is not sufficiently wide.

Example 7.3.2 If (7.3.4) is applied to the matrix A in Example 7.3.1, with $Q_0 = [e_1, e_2]$, we find:

k	$\text{dist}(D_2(A), \text{ran}(Q_k))$
1	.0052
2	.0047
3	.0039
4	.0030
5	.0023
6	.0017
7	.0013

The error is tending to zero with rate $(\lambda_3/\lambda_2)^k = (3/4)^k$.

It is possible to accelerate the convergence in orthogonal iteration using a technique described in Stewart (1976). In the accelerated scheme, the approximate eigenvalue $\lambda_i^{(k)}$ satisfies

$$|\lambda_i^{(k)} - \lambda_i| \approx \left|\frac{\lambda_{r+1}}{\lambda_i}\right|^k \qquad i = 1{:}r\,.$$

(Without the acceleration, the right-hand side is $|\lambda_{i+1}/\lambda_i|^k$.) Stewart's algorithm involves computing the Schur decomposition of the matrices $Q_k^T A Q_k$ every so often. The method can be very useful in situations where A is large and sparse and a few of its largest eigenvalues are required.

7.3.3 The QR Iteration

We now "derive" the QR iteration (7.3.1) and examine its convergence. Suppose $r = n$ in (7.3.4) and the eigenvalues of A satisfy

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|.$$

Partition the matrix Q in (7.3.5) and Q_k in (7.3.4) as follows:

$$Q = [\, q_1, \ldots, q_n \,] \qquad Q_k = \left[\, q_1^{(k)}, \ldots, q_n^{(k)} \,\right]$$

If

$$\text{dist}(D_i(A^H), \text{span}\{q_1^{(0)}, \ldots, q_i^{(0)}\}) \;<\; 1 \qquad i = 1{:}n \qquad (7.3.7)$$

then it follows from Theorem 7.3.1 that

$$\text{dist}(\text{span}\{q_1^{(k)}, \ldots, q_i^{(k)}\}, \text{span}\{q_1, \ldots, q_i\}) \;\to\; 0$$

for $i = 1{:}n$. This implies that the matrices T_k defined by

$$T_k \;=\; Q_k^H A Q_k$$

are converging to upper triangular form. Thus, it can be said that the method of orthogonal iteration computes a Schur decomposition provided the original iterate $Q_0 \in \mathbb{C}^{n\times n}$ is not deficient in the sense of (7.3.7).

The QR iteration arises naturally by considering how to compute the matrix T_k directly from its predecessor T_{k-1}. On the one hand, we have from (7.3.4) and the definition of T_{k-1} that

$$T_{k-1} = Q_{k-1}^H A Q_{k-1} = Q_{k-1}^H (A Q_{k-1}) = (Q_{k-1}^H Q_k) R_k.$$

On the other hand

$$T_k = Q_k^H A Q_k = (Q_k^H A Q_{k-1})(Q_{k-1}^H Q_k) = R_k (Q_{k-1}^H Q_k).$$

Thus, T_k is determined by computing the QR factorization of T_{k-1} and then multiplying the factors together in reverse order. This is precisely what is done in (7.3.1).

Example 7.3.3 If the iteration:

$$\begin{array}{l} \textbf{for } k = 1,2,\ldots \\ \quad A = QR \\ \quad A = RQ \\ \textbf{end} \end{array}$$

is applied to the matrix of Example 7.3.1, then the strictly lower triangular elements diminish as follows:

k	$O(\|a_{21}\|)$	$O(\|a_{31}\|)$	$O(\|a_{32}\|)$
1	10^{-1}	10^{-1}	10^{-2}
2	10^{-2}	10^{-2}	10^{-3}
3	10^{-2}	10^{-3}	10^{-3}
4	10^{-3}	10^{-3}	10^{-3}
5	10^{-3}	10^{-4}	10^{-3}
6	10^{-4}	10^{-5}	10^{-3}
7	10^{-4}	10^{-5}	10^{-3}
8	10^{-5}	10^{-6}	10^{-4}
9	10^{-5}	10^{-7}	10^{-4}
10	10^{-6}	10^{-8}	10^{-4}

Note that a single QR iteration is an $O(n^3)$ calculation. Moreover, since convergence is only linear (when it exists), it is clear that the method is a prohibitively expensive way to compute Schur decompositions. Fortunately these practical difficulties can be overcome, as we show in §7.4 and §7.5.

7.3.4 LR Iterations

We conclude with some remarks about power iterations that rely on the LU factorization rather than the QR factorizaton. Let $G_0 \in \mathbb{C}^{n\times r}$ have rank r. Corresponding to (7.3.4) we have the following iteration:

$$\begin{array}{ll} \textbf{for } k = 1,2,\ldots & \\ \quad Z_k = AG_{k-1} & \\ \quad Z_k = G_k R_k & \text{(LU factorization)} \\ \textbf{end} & \end{array} \tag{7.3.8}$$

Suppose $r = n$ and that we define the matrices T_k by

$$T_k \;=\; G_k^{-1} A G_k. \tag{7.3.9}$$

It can be shown that if we set $L_0 = G_0$, then the T_k can be generated as follows:

$$
\begin{aligned}
& T_0 = L_0^{-1} A L_0 \\
& \textbf{for } k = 1,2,\ldots \\
& \quad T_{k-1} = L_k R_k \qquad \text{(LU factorization)} \\
& \quad T_k = R_k L_k \\
& \textbf{end}
\end{aligned} \tag{7.3.10}
$$

Iterations (7.3.8) and (7.3.10) are known as *treppeniteration* and the *LR iteration,* respectively. Under reasonable assumptions, the T_k converge to upper triangular form. To successfully implement either method, it is necessary to pivot. See Wilkinson (1965, p.602).

Appendix

In order to establish Theorem 7.3.1 we need the following lemma which is concerned with bounding the powers of a matrix and its inverse.

Lemma 7.3.2 *Let $Q^H AQ = T = D + N$ be a Schur decomposition of $A \in \mathbb{C}^{n\times n}$ where D is diagonal and N strictly upper triangular. Let λ and μ denote the largest and smallest eigenvalues of A in absolute value. If $\theta \geq 0$ then for all $k \geq 0$ we have*

$$
\| A^k \|_2 \;\leq\; (1+\theta)^{n-1} \left(|\,\lambda\,| \;+\; \frac{\| N \|_F}{1+\theta} \right)^k . \tag{7.3.11}
$$

If A is nonsingular and $\theta \geq 0$ satisfies $(1+\theta)|\mu| > \| N \|_F$, then for all $k \geq 0$ we also have

$$
\| A^{-k} \|_2 \;\leq\; (1+\theta)^{n-1} \left(\frac{1}{|\mu| - \| N \|_F/(1+\theta)} \right)^k . \tag{7.3.12}
$$

Proof. For $\theta \geq 0$, define the diagonal matrix Δ by

$$
\Delta \;=\; \text{diag}\,(1, (1+\theta), (1+\theta)^2, \ldots, (1+\theta)^{n-1})
$$

and note that $\kappa_2(\Delta) = (1+\theta)^{n-1}$. Since N is strictly upper triangular, it is easy to verify that $\| \Delta N \Delta^{-1} \|_F \leq \| N \|_F/(1+\theta)$. Thus,

$$
\begin{aligned}
\| A^k \|_2 \;&=\; \| T^k \|_2 \;=\; \| \Delta^{-1}(D + \Delta N \Delta^{-1})^k \Delta \|_2 \\
&\leq\; \kappa_2(\Delta) \left(\| D \|_2 + \| \Delta N \Delta^{-1} \|_2 \right)^k \\
&\leq\; (1+\theta)^{n-1} \left(|\lambda| + \frac{\| N \|_F}{1+\theta} \right)^k .
\end{aligned}
$$

On the other hand, if A is nonsingular and $(1+\theta)|\mu| > \| N \|_F$, then $\| \Delta D^{-1} N \Delta^{-1} \|_2 < 1$ and using Lemma 2.3.3 we obtain

$$
\| A^{-k} \|_2 \;=\; \| T^{-k} \|_2 \;=\; \left\| \Delta^{-1} [(I + \Delta D^{-1} N \Delta^{-1})^{-1} D^{-1}]^k \Delta \right\|_2
$$

$$\leq \kappa_2(\Delta)\left(\frac{\| D^{-1} \|_2}{1 - \| \Delta D^{-1} N \Delta^{-1} \|_2}\right)^k$$
$$\leq (1+\theta)^{n-1}\left(\frac{1}{|\mu| - \| N \|_F/(1+\theta)}\right)^k . \square$$

Proof of Theorem 7.3.1

It is easy to show by induction that $A^k Q_0 = Q_k(R_k \cdots R_1)$. By substituting (7.3.5) and 7.3.6) into this equality we obtain

$$T^k \begin{bmatrix} V_0 \\ W_0 \end{bmatrix} = \begin{bmatrix} V_k \\ W_k \end{bmatrix} (R_k \cdots R_1)$$

where $V_k = Q_\alpha^H Q_k$ and $W_k = Q_\beta^H Q_k$. Using Lemma 7.1.5 we know that a matrix $X \in \mathbb{C}^{r \times (n-r)}$ exists such that

$$\begin{bmatrix} I_r & X \\ 0 & I_{n-r} \end{bmatrix}^{-1} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} I_r & X \\ 0 & I_{n-r} \end{bmatrix} = \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix}$$

and so

$$\begin{bmatrix} T_{11}^k & 0 \\ 0 & T_{22}^k \end{bmatrix} \begin{bmatrix} V_0 - XW_0 \\ W_0 \end{bmatrix} = \begin{bmatrix} V_k - XW_k \\ W_k \end{bmatrix} (R_k \cdots R_1) .$$

Below we establish that the matrix $V_0 - XW_0$ is nonsingular and this enables us to obtain the following expression:

$$W_k = T_{22}^k W_0 (V_0 - XW_0)^{-1} T_{11}^{-k} [\, I_r \, , \, -X \,] \begin{bmatrix} V_k \\ W_k \end{bmatrix} .$$

Recalling the definition of distance between subspaces from §2.6.3,

$$\text{dist}(D_r(A), \text{ran}(Q_k)) = \| Q_\beta^H Q_k \|_2 = \| W_k \|_2 .$$

Since

$$\| \, [\, I_r \, , \, -X \,] \, \|_2 \leq 1 + \| X \|_F$$

we have

$$\text{dist}(D_r(A), \text{ran}(Q_k)) \leq \tag{7.3.13}$$
$$\| T_{22}^k \|_2 \, \| (V_0 - XW_0)^{-1} \|_2 \, \| T_{11}^{-k} \|_2 \, (1 + \| X \|_F) .$$

To prove the theorem we must look at each of the four factors in the upper bound.

Since $\text{sep}(T_{11}, T_{22})$ is the smallest singular value of the linear transformation $\phi(X) = T_{11}X - XT_{22}$ it readily follows from $\phi(X) = -T_{12}$ that

$$\| X \|_F \leq \frac{\| T_{12} \|_F}{\text{sep}(T_{11}, T_{22})} . \tag{7.3.14}$$

Using Lemma 7.3.2, it can be shown that

$$\| T_{22}^k \|_2 \leq (1+\theta)^{n-r-1} \left(|\lambda_{r+1}| + \frac{\| N \|_F}{1+\theta} \right)^k \tag{7.3.15}$$

and

$$\| T_{11}^{-k} \|_2 \leq (1+\theta)^{r-1} / \left(|\lambda_r| - \frac{\| N \|_F}{1+\theta} \right)^k . \tag{7.3.16}$$

Finally, we turn our attention to the $\| (V_0 - XW_0)^{-1} \|$ factor. Note that

$$\begin{aligned} V_0 - XW_0 &= Q_\alpha^H Q_0 - X Q_\beta^H Q_0 \\ &= [\, I_r \,,\, -X \,] \begin{bmatrix} Q_\alpha^H \\ Q_\beta^H \end{bmatrix} Q_0 \\ &= \left[[\, Q_\alpha \, Q_\beta \,] \begin{bmatrix} I_r \\ -X^H \end{bmatrix} \right]^H Q_0 \\ &= (I_r + XX^H)^{1/2} (Z^H Q_0) \end{aligned}$$

where

$$\begin{aligned} Z &= [\, Q_\alpha \, Q_\beta \,] \begin{bmatrix} I_r \\ -X^H \end{bmatrix} (I_r + XX^H)^{-1/2} \\ &= (Q_\alpha - Q_\beta X^H)(I_r + XX^H)^{-1/2}. \end{aligned}$$

The columns of this matrix are orthonormal. They are also a basis for $D_r(A^H)$ because

$$A^H (Q_\alpha - Q_\beta X^H) = (Q_\alpha - Q_\beta X^H) T_{11}^H.$$

This last fact follows from the equation $A^H Q = Q T^H$.

From Theorem 2.6.1

$$d = \text{dist}(D_r(A^H), \text{range}(Q_0)) = \sqrt{1 - \sigma_r(Z^H Q_0)^2}$$

and since $d < 1$ by hypothesis,

$$\sigma_r(Z^H Q_0) > 0.$$

This shows that

$$(V_0 - XW_0) = (I_r + XX^H)^{1/2} (Z^H Q_0)$$

is nonsingular and thus,

$$\begin{aligned} \| (V_0 - XW_0)^{-1} \|_2 &\leq \| (I_r + XX^H)^{-1/2} \|_2 \| (Z^H Q_0)^{-1} \|_2 \\ &\leq 1/\sqrt{1-d^2}. \end{aligned} \tag{7.3.17}$$

The theorem follows by substituting (7.3.14)-(7.3.17) into (7.3.13). □

Problems

P7.3.1 (a) Show that if $X \in \mathbb{C}^{n\times n}$ is nonsingular, then $\| A \|_X = \| X^{-1}AX \|_2$ defines a matrix norm with the property that $\| AB \|_X \le \| A \|_X \| B \|_X$. (b) Let $A \in \mathbb{C}^{n\times n}$ and set $\rho = \max |\lambda_i|$. Show that for any $\epsilon > 0$ there exists a nonsingular $X \in \mathbb{C}^{n\times n}$ such that $\| A \|_X = \| X^{-1}AX \|_2 \le \rho + \epsilon$. Conclude that there is a constant M such that $\| A^k \|_2 \le M(\rho + \epsilon)^k$ for all non-negative integers k. (Hint: Set $X = Q\ \mathrm{diag}(1, a, \ldots, a^{n-1})$ where $Q^H AQ = D + N$ is A's Schur decomposition.)

P7.3.2 Verify that (7.3.10) calculates the matrices T_k defined by (7.3.9).

P7.3.3 Suppose $A \in \mathbb{C}^{n\times n}$ is nonsingular and that $Q_0 \in \mathbb{C}^{n\times p}$ has orthonormal columns. The following iteration is referred to as *inverse orthogonal iteration*.

for $k = 1, 2, \ldots$
  Solve $AZ_k = Q_{k-1}$ for $Z_k \in \mathbb{C}^{n\times p}$
  $Z_k = Q_k R_k$ (QR factorization)
end

Explain why this iteration can usually be used to compute the p smallest eigenvalues of A in absolute value. Note that to implement this iteration it is necessary to be able to solve linear systems that involve A. When $p = 1$, the method is referred to as the *inverse power method*.

P7.3.4 Assume $A \in \mathbb{R}^{n\times n}$ has eigenvalues $\lambda_1, \ldots, \lambda_n$ that satisfy

$$\lambda = \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 > |\lambda_5| \ge \cdots \ge |\lambda_n|$$

where λ is positive. Assume that A has two Jordan blocks of the form.

$$\begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}.$$

Discuss the convergence properties of the power method when applied to this matrix. Discuss how the convergence might be accelerated.

Notes and References for Sec. 7.3

A detailed, practical discussion of the power method is given in Wilkinson (1965, chapter 10). Methods are discussed for accelerating the basic iteration, for calculating nondominant eigenvalues, and for handling complex conjugate eigenvalue pairs. The connections among the various power iterations are discussed in

B.N. Parlett and W.G. Poole (1973). "A Geometric Theory for the QR, LU, and Power Iterations," *SIAM J. Num. Anal. 10*, 389–412.

The QR iteration was concurrently developed in

J.G.F. Francis (1961). "The QR Transformation: A Unitary Analogue to the LR Transformation," *Comp. J. 4*, 265–71, 332 34.

V.N. Kublanovskaya (1961). "On Some Algorithms for the Solution of the Complete Eigenvalue Problem," *USSR Comp. Math. Phys. 3*, 637–57.

As can be deduced from the title of the first paper, the LR iteration predates the QR iteration. The former very fundamental algorithm was proposed by

H. Rutishauser (1958). "Solution of Eigenvalue Problems with the LR Transformation," *Nat. Bur. Stand. App. Math. Ser. 49*, 47–81.
B.N. Parlett (1995). "The New qd Algorithms," *ACTA Numerica 5*, 459–491.

Numerous papers on the convergence of the QR iteration have appeared. Several of these are

J.H. Wilkinson (1965). "Convergence of the LR, QR, and Related Algorithms," *Comp. J. 8*, 77–84.
B.N. Parlett (1965). "Convergence of the Q-R Algorithm," *Numer. Math. 7*, 187–93. (Correction in *Numer. Math. 10*, 163–64.)
B.N. Parlett (1966). "Singular and Invariant Matrices Under the QR Algorithm," *Math. Comp. 20*, 611–15.
B.N. Parlett (1968). "Global Convergence of the Basic QR Algorithm on Hessenberg Matrices," *Math. Comp. 22*, 803–17.

Wilkinson (AEP, chapter 9) also discusses the convergence theory for this important algorithm.

Deeper insight into the convergence of the QR algorithm and its connection to other important algorithms can be attained by reading

D.S. Watkins (1982). "Understanding the QR Algorithm," *SIAM Review 24*, 427 440.
T. Nanda (1985). "Differential Equations and the QR Algorithm," *SIAM J. Numer. Anal. 22*, 310–321.
D.S. Watkins (1993). "Some Perspectives on the Eigenvalue Problem," *SIAM Review 35*, 430–471.

The following papers are concerned with various practical and theoretical aspects of simultaneous iteration:

H. Rutishauser (1970). "Simultaneous Iteration Method for Symmetric Matrices," *Numer. Math. 16*, 205 23. See also (Wilkinson and Reinsch(1971, pp. 284–302.
M. Clint and A. Jennings (1971). "A Simultaneous Iteration Method for the Unsymmetric Eigenvalue Problem," *J. Inst. Math. Applic. 8*, 111–21.
A. Jennings and D.R.L. Orr (1971). "Application of the Simultaneous Iteration Method to Undamped Vibration Problems," *Inst. J. Numer. Math. Eng. 3*, 13–24.
A. Jennings and W.J. Stewart (1975). "Simultaneous Iteration for the Partial Eigensolution of Real Matrices," *J. Inst. Math. Applic. 15*, 351–62.
G.W. Stewart (1975). "Methods of Simultaneous Iteration for Calculating Eigenvectors of Matrices," in *Topics in Numerical Analysis II* , ed. John J.H. Miller, Academic Press, New York, pp. 185–96.
G.W. Stewart (1976). "Simultaneous Iteration for Computing Invariant Subspaces of Non-Hermitian Matrices," *Numer. Math. 25*, 123–36.

See also chapter 10 of

A. Jennings (1977). *Matrix Computation for Engineers and Scientists*, John Wiley and Sons, New York.

Simultaneous iteration and the Lanczos algorithm (cf. Chapter 9) are the principal methods for finding a few eigenvalues of a general sparse matrix.

7.4 The Hessenberg and Real Schur Forms

In this and the next section we show how to make the QR iteration (7.3.1) a fast, effective method for computing Schur decompositions. Because the majority of eigenvalue/invariant subspace problems involve real data, we concentrate on developing the real analog of (7.3.1) which we write as follows:

$$
\begin{array}{l}
H_0 = U_0^T A U_0 \\
\textbf{for } k = 1, 2, \ldots \\
\quad H_{k-1} = U_k R_k \qquad \text{(QR factorization)} \\
\quad H_k = R_k U_k \\
\textbf{end}
\end{array}
\tag{7.4.1}
$$

Here, $A \in \mathbb{R}^{n \times n}$, each $U_k \in \mathbb{R}^{n \times n}$ is orthogonal, and each $R_k \in \mathbb{R}^{n \times n}$ is upper triangular. A difficulty associated with this real iteration is that the H_k can never converge to strict, "eigenvalue revealing," triangular form in the event that A has complex eigenvalues. For this reason, we must lower our expectations and be content with the calculation of an alternative decomposition known as the *real Schur decomposition.*

In order to compute the real Schur decomposition efficiently we must carefully choose the initial orthogonal similarity transformation U_0 in (7.4.1). In particular, if we choose U_0 so that H_0 is upper Hessenberg, then the amount of work per iteration is reduced from $O(n^3)$ to $O(n^2)$. The initial reduction to Hessenberg form (the U_0 computation) is a very important computation in its own right and can be realized by a sequence of Householder matrix operations.

7.4.1 The Real Schur Decomposition

A block upper triangular matrix with either 1-by-1 or 2-by-2 diagonal blocks is upper *quasi-triangular.* The real Schur decomposition amounts to a real reduction to upper quasi-triangular form.

Theorem 7.4.1 (Real Schur Decomposition) *If $A \in \mathbb{R}^{n \times n}$, then there exists an orthogonal $Q \in \mathbb{R}^{n \times n}$ such that*

$$
Q^T A Q = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ 0 & R_{22} & \cdots & R_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{mm} \end{bmatrix} \tag{7.4.2}
$$

where each R_{ii} is either a 1-by-1 matrix or a 2-by-2 matrix having complex conjugate eigenvalues.

Proof. The complex eigenvalues of A must come in conjugate pairs, since the characteristic polynomial $\det(zI - A)$ has real coefficients. Let k be

the number of complex conjugate pairs in $\lambda(A)$. We prove the theorem by induction on k. Observe first that Lemma 7.1.2 and Theorem 7.1.3 have obvious real analogs. Thus, the theorem holds if $k = 0$. Now suppose that $k \geq 1$. If $\lambda = \gamma + i\mu \in \lambda(A)$ and $\mu \neq 0$, then there exist vectors y and z in $\mathbb{R}^n (z \neq 0)$ such that $A(y + iz) = (\gamma + i\mu)(y + iz)$, i.e.,

$$A \begin{bmatrix} y & z \end{bmatrix} = \begin{bmatrix} y & z \end{bmatrix} \begin{bmatrix} \gamma & \mu \\ -\mu & \gamma \end{bmatrix}.$$

The assumption that $\mu \neq 0$ implies that y and z span a two-dimensional, real invariant subspace for A. It then follows from Lemma 7.1.2 that an orthogonal $U \in \mathbb{R}^{n \times n}$ exists such that

$$U^T A U = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{matrix} 2 \\ n-2 \end{matrix}$$
$$\qquad\qquad \begin{matrix} 2 & n-2 \end{matrix}$$

where $\lambda(T_{11}) = \{\lambda, \bar{\lambda}\}$. By induction, there exists an orthogonal $\tilde{U}$ so $\tilde{U}^T T_{22} \tilde{U}$ has the required structure. The theorem follows by setting $Q = U$ diag$(I_2, \tilde{U})$. □

The theorem shows that any real matrix is orthogonally similar to an upper quasi-triangular matrix. It is clear that the real and imaginary part of the complex eigenvalues can be easily obtained from the 2-by-2 diagonal blocks.

7.4.2 A Hessenberg QR Step

We now turn our attention to the speedy calculation of a single QR step in (7.4.1). In this regard, the most glaring shortcoming associated with (7.4.1) is that each step requires a full QR factorization costing $O(n^3)$ flops. Fortunately, the amount of work per iteration can be reduced by an order of magnitude if the orthogonal matrix U_0 is judiciously chosen. In particular, if $U_0^T A U_0 = H_0 = (h_{ij})$ is upper Hessenberg ($h_{ij} = 0, i > j + 1$), then each subsequent H_k requires only $O(n^2)$ flops to calculate. To see this we look at the computations $H = QR$ and $H_+ = RQ$ when H is upper Hessenberg. As described in §5.2.4, we can upper triangularize H with a sequence of $n - 1$ Givens rotations: $Q^T H \equiv G_{n-1}^T \cdots G_1^T H = R$. Here, $G_i = G(i, i + 1, \theta_i)$. For the $n = 4$ case there are three Givens premultiplications:

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}$$
$$\longrightarrow \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}.$$

See Algorithm 5.2.3.

The computation $RQ = R(G_1 \cdots G_{n-1})$ is equally easy to implement. In the $n = 4$ case there are three Givens post-multiplications:

$$\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}$$

$$\longrightarrow \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}$$

Overall we obtain the following algorithm:

Algorithm 7.4.1 If H is an n-by-n upper Hessenberg matrix, then this algorithm overwrites H with $H_+ = RQ$ where $H = QR$ is the QR factorization of H.

for $k = 1{:}n-1$
$\quad [\, c(k)\,,\, s(k)\,] = \textbf{givens}(H(k,k), H(k+1,k))$
$$\quad H(k{:}k+1, k{:}n) = \begin{bmatrix} c(k) & s(k) \\ -s(k) & c(k) \end{bmatrix}^T H(k{:}k+1, k{:}n)$$
end
for $k = 1{:}n-1$
$$\quad H(1{:}k+1, k{:}k+1) = H(1{:}k+1, k{:}k+1) \begin{bmatrix} c(k) & s(k) \\ -s(k) & c(k) \end{bmatrix}$$
end

Let $G_k = G(k, k+1, \theta_k)$ be the kth Givens rotation. It is easy to confirm that the matrix $Q = G_1 \cdots G_{n-1}$ is upper Hessenberg. Thus, $RQ = H_+$ is also upper Hessenberg. The algorithm requires about $6n^2$ flops and thus is an order-of-magnitude quicker than a full matrix QR step (7.3.1).

Example 7.4.1 If Algorithm 7.4.1 is applied to:

$$H = \begin{bmatrix} 3 & 1 & 2 \\ 4 & 2 & 3 \\ 0 & .01 & 1 \end{bmatrix},$$

then

$$G_1 = \begin{bmatrix} .6 & -.8 & 0 \\ .8 & .6 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad G_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & .9996 & -.0249 \\ 0 & .0249 & .9996 \end{bmatrix},$$

and

$$H_+ = \begin{bmatrix} 4.7600 & -2.5442 & 5.4653 \\ .3200 & .1856 & -2.1796 \\ .0000 & .0263 & 1.0540 \end{bmatrix}.$$

7.4.3 The Hessenberg Reduction

It remains for us to show how the *Hessenberg decomposition*

$$U_0^T A U_0 = H \qquad U_0^T U_0 = I \tag{7.4.3}$$

can be computed. The transformation U_0 can be computed as a product of Householder matrices $P_1, \ldots, P_{n-2}$. The role of P_k is to zero the kth column below the subdiagonal. In the $n = 6$ case, we have

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_1} \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_2}$$

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_3} \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \end{bmatrix} \xrightarrow{P_4}$$

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

In general, after $k-1$ steps we have computed $k-1$ Householder matrices $P_1, \ldots, P_{k-1}$ such that

$$(P_1 \cdots P_{k-1})^T A (P_1 \cdots P_{k-1}) = \begin{array}{c} \left[\begin{array}{ccc} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ 0 & B_{32} & B_{33} \end{array} \right] \begin{array}{c} k-1 \\ 1 \\ n-k \end{array} \\ \begin{array}{ccc} k-1 & \;\;1\;\; & n-k \end{array} \end{array}$$

is upper Hessenberg through its first $k-1$ columns. Suppose $\bar{P}_k$ is an order $n-k$ Householder matrix such that $\bar{P}_k B_{32}$ is a multiple of $e_1^{(n-k)}$. If $P_k = \text{diag}(I_k, \bar{P}_k)$, then

$$(P_1 \cdots P_k)^T A (P_1 \cdots P_k) = \begin{bmatrix} B_{11} & B_{12} & B_{13}\bar{P}_k \\ B_{21} & B_{22} & B_{23}\bar{P}_k \\ 0 & \bar{P}_k B_{32} & \bar{P}_k B_{33} \bar{P}_k \end{bmatrix}$$

is upper Hessenberg through its first k columns. Repeating this for $k = 1{:}n-2$ we obtain

Algorithm 7.4.2 (Householder Reduction to Hessenberg Form) Given $A \in \mathbb{R}^{n\times n}$, the following algorithm overwrites A with $H = U_0^T A U_0$ where H is upper Hessenberg and U_0 is product of Householder matrices.

```
for k = 1:n − 2
    [v, β] = house(A(k + 1:n, k))
    A(k + 1:n, k:n) = (I − βvv^T)A(k + 1:n, k:n)
    A(1:n, k + 1:n) = A(1:n, k + 1:n)(I − βvv^T)
end
```

This algorithm requires $10n^3/3$ flops. If U_0 is explicitly formed, an additional $4n^3/3$ flops are required. The kth Householder matrix can be represented in $A(k+2{:}n, k)$. See Martin and Wilkinson (1968d) for a detailed description.

The roundoff properties of this method for reducing A to Hessenberg form are very desirable. Wilkinson (1965, p.351) states that the computed Hessenberg matrix $\hat{H}$ satisfies $\hat{H} = Q^T(A+E)Q$, where Q is orthogonal and $\| E \|_F \le cn^2\mathbf{u}\| A \|_F$ with c a small constant.

Example 7.4.2 If

$$A = \begin{bmatrix} 1 & 5 & 7 \\ 3 & 0 & 6 \\ 4 & 3 & 1 \end{bmatrix} \quad \text{and} \quad U_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & .6 & .8 \\ 0 & .8 & -.6 \end{bmatrix}$$

then

$$U_0^T A U_0 = H = \begin{bmatrix} 1.00 & 8.60 & -.20 \\ 5.00 & 4.96 & -.72 \\ 0.00 & 2.28 & -3.96 \end{bmatrix}.$$

7.4.4 Level-3 Aspects

The Hessenberg reduction (Algorithm 7.4.2) is rich in level-2 operations: half gaxpys and half outer product updates. We briefly discuss two methods for introducing level-3 computations into the process.

The first approach involves a block reduction to block Hessenberg form and is quite straightforward. Suppose (for clarity) that $n = rN$ and write

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} \quad . \qquad \begin{matrix} r & n-r \end{matrix}$$

Suppose that we have computed the QR factorization $A_{21} = \bar{Q}_1 R_1$ and that $\bar{Q}_1$ is in WY form. That is, we have $W_1, Y_1 \in \mathbb{R}^{(n-r)\times r}$ such that

$\bar{Q}_1 = I - W_1 Y_1^T$. (See §5.2.2 for details.) If $Q_1 = \text{diag}(I_r, \bar{Q}_1)$ then

$$Q_1^T A Q_1 = \begin{bmatrix} A_{11} & A_{12}\bar{Q}_1 \\ R_1 & \bar{Q}_1^T A_{22} \bar{Q}_1 \end{bmatrix}.$$

Notice that the updates of the (1,2) and (2,2) blocks are rich in level-3 operations given that $\bar{Q}_1$ is in WY form. This fully illustrates the overall process as $Q_1^T A Q_1$ is block upper Hessenberg through its first block column. We next repeat the computations on the first r columns of $\bar{Q}_1^T A_{22} \bar{Q}_1$. After $N-2$ such steps we obtain

$$H = U_0^T A U_0 = \begin{bmatrix} H_{11} & H_{12} & \cdots & \cdots & H_{1N} \\ H_{21} & H_{22} & \cdots & \cdots & H_{2N} \\ 0 & \ddots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & H_{N,N-1} & H_{NN} \end{bmatrix}$$

where each H_{ij} is r-by-r and $U_0 = Q_1 \cdots Q_{N-2}$ with with each Q_i in WY form. The overall algorithm has a level-3 fraction of the form $1 - O(1/N)$.

Note that the subdiagonal blocks in H are upper triangular and so the matrix has lower bandwidth p. It is possible to reduce H to actual Hessenberg form by using Givens rotations to zero all but the first subdiagonal.

Dongarra, Hammarling and Sorensen (1987) have shown how to proceed directly to Hessenberg form using a mixture of gaxpy's and level-3 updates. Their idea involves minimal updating after each Householder transformation is generated. For example, suppose the first Householder P_1 has been computed. To generate P_2 we need just the second column of $P_1 A P_1$, not the full outer product update. To generate P_3 we need just the 3rd column of $P_2 P_1 A P_1 P_2$, etc. In this way, the Householder matrices can be determined using only gaxpy operations. No outer product updates are involved. Once a suitable number of Householder matrices are known they can be aggregated and applied in a level-3 fashion.

7.4.5 Important Hessenberg Matrix Properties

The Hessenberg decomposition is not unique. If Z is any n-by-n orthogonal matrix and we apply Algorithm 7.4.2 to $Z^T A Z$, then $Q^T A Q = H$ is upper Hessenberg where $Q = Z U_0$. However, $Q e_1 = Z(U_0 e_1) = Z e_1$ suggesting that H is unique once the first column of Q is specified. This is essentially the case provided H has no zero subdiagonal entries. Hessenberg matrices with this property are said to be *unreduced*. Here is a very important theorem that clarifies the uniqueness of the Hessenberg reduction.

Theorem 7.4.2 (Implicit Q Theorem) *Suppose* $Q = [\, q_1, \ldots, q_n \,]$ *and* $V = [\, v_1, \ldots, v_n \,]$ *are orthogonal matrices with the property that both* $Q^T A Q$

$= H$ and $V^TAV = G$ are upper Hessenberg where $A \in \mathbb{R}^{n\times n}$. Let k denote the smallest positive integer for which $h_{k+1,k} = 0$, with the convention that $k = n$ if H is unreduced. If $q_1 = v_1$, then $q_i = \pm v_i$ and $|h_{i,i-1}| = |g_{i,i-1}|$ for $i = 2{:}k$. Moreover, if $k < n$, then $g_{k+1,k} = 0$.

Proof. Define the orthogonal matrix $W = [\, w_1, \ldots, w_n \,] = V^TQ$ and observe that $GW = WH$. By comparing column $i-1$ in this equation for $i = 2{:}k$ we see that

$$h_{i,i-1}w_i \;=\; Gw_{i-1} \;-\; \sum_{j=1}^{i-1} h_{j,i-1}w_j\,.$$

Since $w_1 = e_1$, it follows that $[\, w_1, \ldots, w_k \,]$ is upper triangular and thus $w_i = \pm I_n(:,i) = \pm e_i$ for $i = 2{:}k$. Since $w_i = V^Tq_i$ and $h_{i,i-1} = w_i^TGw_{i-1}$ it follows that $v_i = \pm q_i$ and

$$|h_{i,i-1}| = |q_i^TAq_{i-1}| = |v_i^TAv_{i-1}| = |g_{i,i-1}|$$

for $i = 2{:}k$. If $k < n$, then

$$\begin{aligned} g_{k+1,k} &= e_{k+1}^TGe_k \;=\; e_{k+1}^TGWe_k \;=\; e_{k+1}^TWHe_k \\ &= e_{k+1}^T\sum_{i=1}^{k} h_{ik}We_i \;=\; \sum_{i=1}^{k} h_{ik}e_{k+1}^Te_i \;=\; 0. \square \end{aligned}$$

The gist of the implicit Q theorem is that if $Q^TAQ = H$ and $Z^TAZ = G$ are each unreduced upper Hessenberg matrices and Q and Z have the same first column, then G and H are "essentially equal" in the sense that $G = D^{-1}HD$ where $D = \text{diag}(\pm 1, \ldots, \pm 1)$.

Our next theorem involves a new type of matrix called a *Krylov* matrix. If $A \in \mathbb{R}^{n\times n}$ and $v \in \mathbb{R}^n$, then the Krylov matrix $K(A, v, j) \in \mathbb{R}^{n\times j}$ is defined by

$$K(A,v,j) = \left[\, v\,,\; Av\,, \cdots,\; A^{j-1}v \,\right].$$

It turns out that there is a connection between the Hessenberg reduction $Q^TAQ = H$ and the QR factorization of the Krylov matrix $K(A, Q(:,1), n)$.

Theorem 7.4.3 *Suppose $Q \in \mathbb{R}^{n\times n}$ is an orthogonal matrix and $A \in \mathbb{R}^{n\times n}$. Then $Q^TAQ = H$ is an unreduced upper Hessenberg matrix if and only if $Q^TK(A, Q(:,1), n) = R$ is nonsingular and upper triangular.*

Proof. Suppose $Q \in \mathbb{R}^{n\times n}$ is orthogonal and set $H = Q^TAQ$. Consider the identity

$$Q^TK(A,Q(:,1),n) \;=\; \left[\, e_1,\, He_1, \ldots, H^{n-1}e_1 \,\right] \;\equiv\; R\,.$$

If H is an unreduced upper Hessenberg matrix, then it is clear that R is upper triangular with $r_{ii} = h_{21}h_{32}\cdots h_{i,i-1}$ for $i = 2{:}n$. Since $r_{11} = 1$ it follows that R is nonsingular.

To prove the converse, suppose R is upper triangular and nonsingular. Since $R(:,k+1) = HR(:,k)$ it follows that $H(:,k) \in \text{span}\{ e_1, \ldots, e_{k+1} \}$. This implies that H is upper Hessenberg. Since $r_{nn} = h_{21}h_{32}\cdots h_{n,n-1} \neq 0$ it follows that H is also unreduced. □

Thus, there is more or less a correspondence between nonsingular Krylov matrices and orthogonal similarity reductions to unreduced Hessenberg form.

Our last result concerns eigenvalues of an unreduced upper Hessenberg matrix.

Theorem 7.4.4 *If λ is an eigenvalue of an unreduced upper Hessenberg matrix $H \in \mathbb{R}^{n\times n}$, then its geometric multiplicity is one.*

Proof. For any $\lambda \in \mathbb{C}$ we have $\text{rank}(A - \lambda I) \geq n-1$ because the first $n-1$ columns of $H - \lambda I$ are independent. □

7.4.6 Companion Matrix Form

Just as the Schur decomposition has a nonunitary analog in the Jordan decomposition, so does the Hessenberg decomposition have a nonunitary analog in the *companion matrix decomposition.* Let $x \in \mathbb{R}^n$ and suppose that the Krylov matrix $K = K(A, x, n)$ is nonsingular. If $c = c(0{:}n-1)$ solves the linear system $Kc = -A^n x$, then it follows that $AK = KC$ where C has the form:

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & \cdots & 0 & -c_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -c_{n-1} \end{bmatrix}. \tag{7.4.4}$$

The matrix C is said to be a *companion matrix.* Since

$$\det(zI - C) = c_0 + c_1 z + \cdots + c_{n-1}z^{n-1} + z^n$$

it follows that if K is nonsingular, then the decomposition $K^{-1}AK = C$ displays A's characteristic polynomial. This, coupled with the sparseness of C, has led to "companion matrix methods" in various application areas. These techniques typically involve:

- Computing the Hessenberg decomposition $U_0^T A U_0 = H$.
- Hoping H is unreduced and setting $Y = \left[e_1,\ He_1,\ \ldots,\ H^{n-1}e_1\right]$.
- Solving $YC = HY$ for C.

Unfortunately, this calculation can be highly unstable. A is similar to an unreduced Hessenberg matrix only if each eigenvalue has unit geometric multiplicity. Matrices that have this property are called *nonderogatory*. It follows that the matrix Y above can be very poorly conditioned if A is close to a derogatory matrix.

A full discussion of the dangers associated with companion matrix computation can be found in Wilkinson (1965, pp. 405 ff.).

7.4.7 Hessenberg Reduction Via Gauss Transforms

While we are on the subject of nonorthogonal reduction to Hessenberg form, we should mention that Gauss transformations can be used in lieu of Householder matrices in Algorithm 7.4.2. In particular, suppose permutations $\Pi_1, \ldots, \Pi_{k-1}$ and Gauss transformations $M_1, \ldots, M_{k-1}$ have been determined such that

$$(M_{k-1}\Pi_{k-1} \cdots M_1\Pi_1)A(M_{k-1}\Pi_{k-1} \cdots M_1\Pi_1)^{-1} = B$$

where

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ 0 & B_{32} & B_{33} \end{bmatrix} \begin{matrix} k-1 \\ 1 \\ n-k \end{matrix}$$
$$\qquad\qquad k-1 \qquad 1 \qquad n-k$$

is upper Hessenberg through its first $k-1$ columns. A permutation $\bar{\Pi}_k$ of order $n-k$ is then determined such that the first element of $\bar{\Pi}_k B_{32}$ is maximal in absolute value. This makes it possible to determine a stable Gauss transformation $\bar{M}_k = I - z_k e_1^T$ also of order $n-k$, such that all but the first component of $\bar{M}_k(\bar{\Pi}_k B_{32})$ is zero. Defining $\Pi_k = \text{diag}(I_k, \bar{\Pi}_k)$ and $M_k = \text{diag}(I_k, \bar{M}_k)$, we see that

$$(M_k\Pi_k \cdots M_1\Pi_1)A(M_k\Pi_k \cdots M_1\Pi_1)^{-1} =$$
$$\begin{bmatrix} B_{11} & B_{12} & B_{13}\bar{\Pi}_k^T\bar{M}_k^{-1} \\ B_{21} & B_{22} & B_{23}\bar{\Pi}_k^T\bar{M}_k^{-1} \\ 0 & \bar{M}_k\bar{\Pi}_k B_{32} & \bar{M}_k\bar{\Pi}_k B_{33}\bar{\Pi}_k^T\bar{M}_k^{-1} \end{bmatrix}$$

is upper Hessenberg through its first k columns. Note that $\bar{M}_k^{-1} = I + z_k e_1^T$ and so some very simple rank-one updates are involved in the reduction.

A careful operation count reveals that the Gauss reduction to Hessenberg form requires only half the number of flops of the Householder method. However, as in the case of Gaussian elimination with partial pivoting, there is a (fairly remote) chance of 2^n growth. See Businger (1969). Another difficulty associated with the Gauss approach is that the eigenvalue condition

numbers — the $s(\lambda)^{-1}$ — are not preserved with nonorthogonal similarity transformations and this complicates the error estimation process.

Problems

P7.4.1 Suppose $A \in \mathbb{R}^{n\times n}$ and $z \in \mathbb{R}^n$. Give a detailed algorithm for computing an orthogonal Q such that Q^TAQ is upper Hessenberg and Q^Tz is a multiple of e_1. (Hint: Reduce z first and then apply Algorithm 7.4.2.)

P7.4.2 Specify a complete reduction to Hessenberg form using Gauss transformations and verify that it only requires $5n^3/3$ flops.

P7.4.3 In some situations, it is necessary to solve the linear system $(A + zI)x = b$ for many different values of $z \in \mathbb{R}$ and $b \in \mathbb{R}^n$. Show how this problem can be efficiently and stably solved using the Hessenberg decomposition.

P7.4.4 Give a detailed algorithm for explicitly computing the matrix U_0 in Algorithm 7.4.2. Design your algorithm so that H is overwritten by U_0.

P7.4.5 Suppose $H \in \mathbb{R}^{n\times n}$ is an unreduced upper Hessenberg matrix. Show that there exists a diagonal matrix D such that each subdiagonal element of $D^{-1}HD$ is equal to one. What is $\kappa_2(D)$?

P7.4.6 Suppose $W, Y \in \mathbb{R}^{n\times n}$ and define the matrices C and B by

$$C = W + iY, \qquad B = \begin{bmatrix} W & -Y \\ Y & W \end{bmatrix}$$

Show that if $\lambda \in \lambda(C)$ is real, then $\lambda \in \lambda(B)$. Relate the corresponding eigenvectors.

P7.4.7 Suppose $A = \begin{bmatrix} w & x \\ y & z \end{bmatrix}$ is a real matrix having eigenvalues $\lambda \pm i\mu$, where μ is nonzero. Give an algorithm that stably determines $c = \cos(\theta)$ and $s = \sin(\theta)$ such that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} w & x \\ y & z \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} \lambda & \beta \\ \alpha & \lambda \end{bmatrix}$$

where $\alpha\beta = -\mu^2$.

P7.4.8 Suppose (λ, x) is a known eigenvalue-eigenvector pair for the upper Hessenberg matrix $H \in \mathbb{R}^{n\times n}$. Give an algorithm for computing an orthogonal matrix P such that

$$P^THP = \begin{bmatrix} \lambda & w^T \\ 0 & H_1 \end{bmatrix}$$

where $H_1 \in \mathbb{R}^{(n-1)\times(n-1)}$ is upper Hessenberg. Compute P as a product of Givens rotations.

P7.4.9 Suppose $H \in \mathbb{R}^{n\times n}$ has lower bandwidth p. Show how to compute $Q \in \mathbb{R}^{n\times n}$, a product of Givens rotations, such that Q^THQ is upper Hessenberg. How many flops are required?

P7.4.10 Show that if C is a companion matrix with distinct eigenvalues $\lambda_1, \ldots, \lambda_n$, then $VCV^{-1} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ where

$$V = \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^{n-1} \end{bmatrix}.$$

Notes and References for Sec. 7.4

The real Schur decomposition was originally presented in

F.D. Murnaghan and A. Wintner (1931). "A Canonical Form for Real Matrices Under Orthogonal Transformations," *Proc. Nat. Acad. Sci. 17*, 417–20.

A thorough treatment of the reduction to Hessenberg form is given in Wilkinson (1965, chapter 6), and Algol procedures for both the Householder and Gauss methods appear in

R.S. Martin and J.H. Wilkinson (1968). "Similarity Reduction of a General Matrix to Hessenberg Form," *Numer. Math. 12*, 349–68. See also Wilkinson and Reinsch (1971,pp.339–58).

Fortran versions of the Algol procedures in the last reference are in Eispack.

Givens rotations can also be used to compute the Hessenberg decomposition. See

W. Rath (1982). "Fast Givens Rotations for Orthogonal Similarity," *Numer. Math. 40*, 47 56.

The high performance computation of the Hessenberg reduction is discussed in

J.J. Dongarra, L. Kaufman, and S. Hammarling (1986). "Squeezing the Most Out of Eigenvalue Solvers on High Performance Computers," *Lin. Alg. and Its Applic. 77*, 113–136.

J.J. Dongarra, S. Hammarling, and D.C. Sorensen (1989). "Block Reduction of Matrices to Condensed Forms for Eigenvalue Computations," *JACM 27*, 215–227.

M.W. Berry, J.J. Dongarra, and Y. Kim (1995). "A Parallel Algorithm for the Reduction of a Nonsymmetric Matrix to Block Upper Hessenberg Form," *Parallel Computing 21*, 1189–1211.

The possibility of exponential growth in the Gauss transformation approach was first pointed out in

P. Businger (1969). "Reducing a Matrix to Hessenberg Form," *Math. Comp. 23*, 819–21.

However, the algorithm should be regarded in the same light as Gaussian elimination with partial pivoting—stable for all practical purposes. See Eispack, pp. 56–58.

Aspects of the Hessenberg decomposition for sparse matrices are discussed in

I.S. Duff and J.K. Reid (1975). "On the Reduction of Sparse Matrices to Condensed Forms by Similarity Transformations," *J. Inst. Math. Applic. 15*, 217–24.

Once an eigenvalue of an unreduced upper Hessenberg matrix is known, it is possible to zero the last subdiagonal entry using Givens similarity transformations. See

P.A. Businger (1971). "Numerically Stable Deflation of Hessenberg and Symmetric Tridiagonal Matrices,*BIT 11*, 262–70.

Some interesting mathematical properties of the Hessenberg form may be found in

B.N. Parlett (1967). "Canonical Decomposition of Hessenberg Matrices," *Math. Comp. 21*, 223–27.

Y. Ikebe (1979). "On Inverses of Hessenberg Matrices," *Lin. Alg. and Its Applic. 24*, 93–97.

Although the Hessenberg decomposition is largely appreciated as a "front end" decomposition for the QR iteration, it is increasingly popular as a cheap alternative to the more expensive Schur decomposition in certain problems. For a sampling of applications where it has proven to be very useful, consult

W. Enright (1979). "On the Efficient and Reliable Numerical Solution of Large Linear Systems of O.D.E.'s," *IEEE Trans. Auto. Cont.* AC-24, 905–8.

G.H. Golub, S. Nash and C. Van Loan (1979). "A Hessenberg-Schur Method for the Problem $AX + XB = C$," *IEEE Trans. Auto. Cont. AC-24*, 909–13.
A. Laub (1981). "Efficient Multivariable Frequency Response Computations," *IEEE Trans. Auto. Cont. AC-26*, 407–8.
C.C. Paige (1981). "Properties of Numerical Algorithms Related to Computing Controllability," *IEEE Trans. Auto. Cont. AC-26*, 130–38.
G. Miminis and C.C. Paige (1982). "An Algorithm for Pole Assignment of Time Invariant Linear Systems," *International J. of Control 35*, 341–354.
C. Van Loan (1982). "Using the Hessenberg Decomposition in Control Theory," in *Algorithms and Theory in Filtering and Control* , D.C. Sorensen and R.J. Wets (eds), Mathematical Programming Study No. 18, North Holland, Amsterdam, pp. 102–11.

The advisability of posing polynomial root problems as companion matrix eigenvalue problem is discussed in

K.-C. Toh and L.N. Trefethen (1994). "Pseudozeros of Polynomials and Pseudospectra of Companion Matrices," *Numer. Math. 68*, 403 425.
A. Edelman and H. Murakami (1995). "Polynomial Roots from Companion Matrix Eigenvalues," *Math. Comp. 64*, 763–776.

7.5 The Practical QR Algorithm

We return to the Hessenberg QR iteration which we write as follows:

$$\begin{array}{ll} H = U_0^T A U_0 & \text{(Hessenberg Reduction)} \\ \textbf{for } k = 1,2,\ldots & \\ \quad H = UR & \text{(QR factorization)} \\ \quad H = RU & \\ \textbf{end} & \end{array} \tag{7.5.1}$$

Our aim in this section is to describe how the H's converge to upper quasi-triangular form and to show how the convergence rate can be accelerated by incorporating *shifts*.

7.5.1 Deflation

Without loss of generality we may assume that each Hessenberg matrix H in (7.5.1) is unreduced. If not, then at some stage we have

$$H = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix}$$
$$\qquad\quad p \quad n-p$$

where $1 \le p < n$ and the problem *decouples* into two smaller problems involving H_{11} and H_{22}. The term *deflation* is also used in this context, usually when $p = n - 1$ or $n - 2$.

In practice, decoupling occurs whenever a subdiagonal entry in H is suitably small. For example, in Eispack if

$$|h_{p+1,p}| \le c\mathbf{u}(|h_{pp}| + |h_{p+1,p+1}|) \tag{7.5.2}$$

for a small constant c, then $h_{p+1,p}$ is "declared" to be zero. This is justified since rounding errors of order $\mathbf{u}\| H \|$ are already present throughout the matrix.

7.5.2 The Shifted QR Iteration

Let $\mu \in \mathbb{R}$ and consider the iteration:

$$
\begin{array}{ll}
H = U_0^T A U_0 & \text{(Hessenberg Reduction)} \\
\textbf{for } k = 1, 2, \ldots & \\
\quad \text{Determine a scalar } \mu. & \\
\quad H - \mu I = UR & \text{(QR factorization)} \\
\quad H = RU + \mu I & \\
\textbf{end} &
\end{array}
\tag{7.5.3}
$$

The scalar μ is referred to as a *shift* . Each matrix H generated in (7.5.3) is similar to A, since $RU + \mu I = U^T(UR + \mu I)U = U^T H U$. If we order the eigenvalues λ_i of A so that

$$|\lambda_1 - \mu| \geq \cdots \geq |\lambda_n - \mu|,$$

and μ is fixed from iteration to iteration, then the theory of §7.3 says that the pth subdiagonal entry in H converges to zero with rate

$$\left| \frac{\lambda_{p+1} - \mu}{\lambda_p - \mu} \right|^k .$$

Of course, if $\lambda_p = \lambda_{p+1}$, then there is no convergence at all. But if, for example, μ is much closer to λ_n than to the other eigenvalues, then the zeroing of the $(n, n-1)$ entry is rapid. In the extreme case we have the following:

Theorem 7.5.1 *Let μ be an eigenvalue of an n-by-n unreduced Hessenberg matrix H. If $\bar{H} = RU + \mu I$, where $H - \mu I = UR$ is the QR factorization of $H - \mu I$, then $\bar{h}_{n,n-1} = 0$ and $\bar{h}_{nn} = \mu$.*

Proof. Since H is an unreduced Hessenberg matrix the first $n-1$ columns of $H - \mu I$ are independent, regardless of μ. Thus, if $UR = (H - \mu I)$ is the QR factorization then $r_{ii} \neq 0$ for $i = 1{:}n-1$. But if $H - \mu I$ is singular then $r_{11} \cdots r_{nn} = 0$. Thus, $r_{nn} = 0$ and $\bar{H}(n,:) = [\,0,\ \ldots,\ 0,\ \mu\,]$. □

The theorem says that if we shift by an exact eigenvalue, then in exact arithmetic deflation occurs in one step.

Example 7.5.1 If

$$H = \begin{bmatrix} 9 & -1 & -2 \\ 2 & 6 & -2 \\ 0 & 1 & 5 \end{bmatrix},$$

then $6 \in \lambda(H)$. If $UR = H - 6I$ is the QR factorization, then $\bar{H} = RU + 6I$ is given by

$$\bar{H} = \begin{bmatrix} 8.5384 & -3.7313 & -1.0090 \\ 0.6343 & 5.4615 & 1.3867 \\ 0.0000 & 0.0000 & 6.0000 \end{bmatrix}.$$

7.5.3 The Single Shift Strategy

Now let us consider varying μ from iteration to iteration incorporating new information about $\lambda(A)$ as the subdiagonal entries converge to zero. A good heuristic is to regard h_{nn} as the best approximate eigenvalue along the diagonal. If we shift by this quantity during each iteration, we obtain the *single-shift QR iteration*:

$$\begin{array}{l} \textbf{for } k = 1,2,\ldots \\ \quad \mu = H(n,n) \\ \quad H - \mu I = UR \qquad \text{(QR Factorization)} \\ \quad H = RU + \mu I \\ \textbf{end} \end{array} \qquad (7.5.4)$$

If the $(n, n-1)$ entry converges to zero, it is likely to do so at a quadratic rate. To see this, we borrow an example from Stewart (1973, p. 366). Suppose H is an unreduced upper Hessenberg matrix of the form

$$H = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \epsilon & h_{nn} \end{bmatrix}$$

and that we perform one step of the single-shift QR algorithm: $UR = H - h_{nn}I$, $\bar{H} = RU + h_{nn}I$. After $n-2$ steps in the reduction of $H - h_{nn}I$ to upper triangular form we obtain a matrix with the following structure:

$$H = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & a & b \\ 0 & 0 & 0 & \epsilon & 0 \end{bmatrix}$$

It is not hard to show that the $(n, n-1)$ entry in $\bar{H} = RU + h_{nn}I$ is given by $-\epsilon^2 b/(\epsilon^2 + a^2)$. If we assume that $\epsilon \ll a$, then it is clear that

the new $(n, n-1)$ entry has order ϵ^2, precisely what we would expect of a quadratically converging algorithm.

Example 7.5.2 If

$$H = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & .001 & 7 \end{bmatrix}$$

and $UR = H - 7I$ is the QR factorization, then $\bar{H} = RU + 7I$ is given by

$$\bar{H} \approx \begin{bmatrix} -0.5384 & 1.6908 & 0.8351 \\ 0.3076 & 6.5264 & -6.6555 \\ 0.0000 & 2 \cdot 10^{-5} & 7.0119 \end{bmatrix}.$$

Near-perfect shifts as above almost always ensure a small $\bar{h}_{n,n-1}$. However, this is just a heuristic. There are examples in which $\bar{h}_{n,n-1}$ is a relatively large matrix entry even though $\sigma_{min}(H - \mu I) \approx \mathbf{u}$.

7.5.4 The Double Shift Strategy

Unfortunately, difficulties with (7.5.4) can be expected if at some stage the eigenvalues a_1 and a_2 of

$$G = \begin{bmatrix} h_{mm} & h_{mn} \\ h_{nm} & h_{nn} \end{bmatrix} \qquad m = n - 1 \tag{7.5.5}$$

are complex for then h_{nn} would tend to be a poor approximate eigenvalue.

A way around this difficulty is to perform two single-shift QR steps in succession using a_1 and a_2 as shifts:

$$\begin{aligned} H - a_1 I &= U_1 R_1 \\ H_1 &= R_1 U_1 + a_1 I \\ H_1 - a_2 I &= U_2 R_2 \\ H_2 &= R_2 U_2 + a_2 I \end{aligned} \tag{7.5.6}$$

These equations can be manipulated to show that

$$(U_1 U_2)(R_2 R_1) = M \tag{7.5.7}$$

where M is defined by

$$M = (H - a_1 I)(H - a_2 I). \tag{7.5.8}$$

Note that M is a real matrix even if G's eigenvalues are complex since

$$M = H^2 - sH + tI$$

where

$$s = a_1 + a_2 = h_{mm} + h_{nn} = \text{trace}(G) \in \mathbb{R}$$

and

$$t = a_1 a_2 = h_{mm}h_{nn} - h_{mn}h_{nm} = \det(G) \in \mathbb{R}\,.$$

Thus, (7.5.7) is the QR factorization of a real matrix and we may choose U_1 and U_2 so that $Z = U_1 U_2$ is real orthogonal. It then follows that

$$H_2 = U_2^H H_1 U_2 = U_2^H (U_1^H H U_1) U_2 = (U_1 U_2)^H H (U_1 U_2) = Z^T H Z.$$

is real.

Unfortunately, roundoff error almost always prevents an exact return to the real field. A real H_2 could be guaranteed if we

- explicitly form the real matrix $M = H^2 - sH + tI$,
- compute the real QR factorization $M = ZR$, and
- set $H_2 = Z^T H Z$.

But since the first of these steps requires $O(n^3)$ flops, this is not a practical course of action.

7.5.5 The Double Implicit Shift Strategy

Fortunately, it turns out that we can implement the double shift step with $O(n^2)$ flops by appealing to the Implicit Q Theorem of §7.4.5. In particular we can effect the transition from H to H_2 in $O(n^2)$ flops if we

- compute Me_1, the first column of M;
- determine a Householder matrix P_0 such that $P_0(Me_1)$ is a multiple of e_1;
- compute Householder matrices $P_1, \ldots, P_{n-2}$ such that if Z_1 is the product $Z_1 = P_0 P_1 \cdots P_{n-2}$, then $Z_1^T H Z_1$ is upper Hessenberg and the first columns of Z and Z_1 are the same.

Under these circumstances, the Implicit Q theorem permits us to conclude that if $Z^T H Z$ and $Z_1^T H Z_1$ are both unreduced upper Hessenberg matrices, then they are essentially equal. Note that if these Hessenberg matrices are not unreduced, then we can effect a decoupling and proceed with smaller unreduced subproblems.

Let us work out the details. Observe first that P_0 can be determined in $O(1)$ flops since $Me_1 = [x,\ y,\ z,\ 0, \ldots, 0]^T$ where

$$\begin{aligned} x &= h_{11}^2 + h_{12}h_{21} - sh_{11} + t \\ y &= h_{21}(h_{11} + h_{22} - s) \\ z &= h_{21}h_{32}. \end{aligned}$$

Since a similarity transformation with P_0 only changes rows and columns 1, 2, and 3, we see that

$$P_0HP_0 = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Now the mission of the Householder matrices $P_1, \ldots, P_{n-2}$ is to restore this matrix to upper Hessenberg form. The calculation proceeds as follows:

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{P_1} \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{P_2}$$

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_3} \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \end{bmatrix} \xrightarrow{P_4}$$

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Clearly, the general P_k has the form $P_k = \text{diag}(I_k, \tilde{P}_k, I_{n-k-3})$ where $\tilde{P}_k$ is a 3-by-3 Householder matrix. For example,

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that P_{n-2} is an exception to this since $P_{n-2} = \text{diag}(I_{n-2}, \tilde{P}_{n-2})$.

The applicability of Theorem 7.4.3 (the Implicit Q theorem) follows from the observation that $P_ke_1 = e_1$ for $k = 1{:}n-2$ and that P_0 and Z

have the same first column. Hence, $Z_1e_1 = Ze_1$, and we can assert that Z_1 essentially equals Z provided that the upper Hessenberg matrices Z^THZ and $Z_1^THZ_1$ are each unreduced.

The implicit determination of H_2 from H outlined above was first described by Francis (1961) and we refer to it as a *Francis QR step*. The complete Francis step is summarized as follows:

Algorithm 7.5.1 (Francis QR Step) Given the unreduced upper Hessenberg matrix $H \in \mathbb{R}^{n\times n}$ whose trailing 2-by-2 principal submatrix has eigenvalues a_1 and a_2, this algorithm overwrites H with Z^THZ, where $Z = P_1 \cdots P_{n-2}$ is a product of Householder matrices and $Z^T(H - a_1I)(H - a_2I)$ is upper triangular.

```
m = n − 1
{Compute first column of (H − a1 I)(H − a2 I).}
s = H(m, m) + H(n, n)
t = H(m, m)H(n, n) − H(m, n)H(n, m)
x = H(1, 1)H(1, 1) + H(1, 2)H(2, 1) − sH(1, 1) + t
y = H(2, 1)(H(1, 1) + H(2, 2) − s)
z = H(2, 1)H(3, 2)
for k = 0:n − 3
    [v, β] = house([x y z]^T)
    q = max{1, k}.
    H(k + 1:k + 3, q:n) = (I − βvv^T)H(k + 1:k + 3, q:n)
    r = min{k + 4, n}
    H(1:r, k + 1:k + 3) = H(1:r, k + 1:k + 3)(I − βvv^T)
    x = H(k + 2, k + 1)
    y = H(k + 3, k + 1)
    if k < n − 3
        z = H(k + 4, k + 1)
    end
end
[v, β] = house([ x y ]^T)
H(n − 1:n, n − 2:n) = (I − βvv^T)H(n − 1:n, n − 2:n)
H(1:n, n − 1:n) = H(1:n, n − 1:n)(I − βvv^T)
```

This algorithm requires $10n^2$ flops. If Z is accumulated into a given orthogonal matrix, an additional $10n^2$ flops are necessary.

7.5.6 The Overall Process

Reducing A to Hessenberg form using Algorithm 7.4.2 and then iterating with Algorithm 7.5.1 to produce the real Schur form is the standard means by which the dense unsymmetric eigenproblem is solved. During the iteration it is necessary to monitor the subdiagonal elements in H in order to

spot any possible decoupling. How this is done is illustrated in the following algorithm:

Algorithm 7.5.2 (QR Algorithm) Given $A \in \mathbb{R}^{n\times n}$ and a tolerance *tol* greater than the unit roundoff, this algorithm computes the real Schur canonical form $Q^T AQ = T$. A is overwritten with the Hessenberg decomposition. If Q and T are desired, then T is stored in H. If only the eigenvalues are desired, then diagonal blocks in T are stored in the corresponding positions in H.

Use Algorithm 7.4.2 to compute the Hessenberg reduction
    $H = U_0^T A U_0$ where $U_0 = P_1 \cdots P_{n-2}$.
If Q is desired form $Q = P_1 \cdots P_{n-2}$. See§5.1.6.
until $q = n$
    Set to zero all subdiagonal elements that satisfy:
        $|h_{i,i-1}| \leq tol(|h_{ii}| + |h_{i-1,i-1}|)$.
    Find the largest non-negative q and the smallest
        non-negative p such that

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ 0 & H_{22} & H_{23} \\ 0 & 0 & H_{33} \end{bmatrix} \begin{matrix} p \\ n-p-q \\ q \end{matrix}$$
$$\begin{matrix} \quad p & \quad n-p-q & \quad q \end{matrix}$$

        where H_{33} is upper quasi-triangular and H_{22} is
        unreduced. (Note: either p or q may be zero.)
    if $q < n$
        Perform a Francis QR step on H_{22}: $H_{22} = Z^T H_{22} Z$
        if Q is desired
            $Q = Q\text{diag}(I_p, Z, I_q)$
            $H_{12} = H_{12} Z$
            $H_{23} = Z^T H_{23}$
        end
    end
end
Upper triangularize all 2-by-2 diagonal blocks in H that have
    real eigenvalues and accumulate the transformations
    if necessary.

This algorithm requires $25n^3$ flops if Q and T are computed. If only the eigenvalues are desired, then $10n^3$ flops are necessary. These flops counts are very approximate and are based on the empirical observation that on average only two Francis iterations are required before the lower 1-by-1 or

2-by-2 decouples.

Example 7.5.3 If Algorithm 7.5.2 is applied to

$$A = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 4 & 4 & 5 & 6 & 7 \\ 0 & 3 & 6 & 7 & 8 \\ 0 & 0 & 2 & 8 & 9 \\ 0 & 0 & 0 & 1 & 10 \end{bmatrix},$$

then the subdiagonal entries converge as follows

Iteration	$O(\lvert h_{21}\rvert)$	$O(\lvert h_{32}\rvert)$	$O(\lvert h_{43}\rvert)$	$O(\lvert h_{54}\rvert)$
1	10^0	10^0	10^0	10^0
2	10^0	10^0	10^0	10^0
3	10^0	10^0	10^{-1}	10^0
4	10^0	10^0	10^{-3}	10^{-3}
5	10^0	10^0	10^{-6}	10^{-5}
6	10^{-1}	10^0	10^{-13}	10^{-13}
7	10^{-1}	10^0	10^{-28}	10^{-13}
8	10^{-4}	10^0	converg.	converg.
9	10^{-8}	10^0		
10	10^{-8}	10^0		
11	10^{-16}	10^0		
12	10^{-32}	10^0		
13	converg.	converg.		

The roundoff properties of the QR algorithm are what one would expect of any orthogonal matrix technique. The computed real Schur form $\hat{T}$ is orthogonally similar to a matrix near to A, i.e.,

$$Q^T(A+E)Q = \hat{T}$$

where $Q^TQ = I$ and $\| E \|_2 \approx \mathbf{u}\| A \|_2$. The computed $\hat{Q}$ is almost orthogonal in the sense that $\hat{Q}^T\hat{Q} = I + F$ where $\| F \|_2 \approx \mathbf{u}$.

The order of the eigenvalues along $\hat{T}$ is somewhat arbitrary. But as we discuss in §7.6, any ordering can be achieved by using a simple procedure for swapping two adjacent diagonal entries.

7.5.7 Balancing

Finally, we mention that if the elements of A have widely varying magnitudes, then A should be *balanced* before applying the QR algorithm. This is an $O(n^2)$ calculation in which a diagonal matrix D is computed so that if

$$D^{-1}AD = [\, c_1, \ldots, c_n \,] = \begin{bmatrix} r_1^T \\ \vdots \\ r_n^T \end{bmatrix}$$

then $\| r_i \|_\infty \approx \| c_i \|_\infty$ for $i = 1{:}n$. The diagonal matrix D is chosen to have the form $D = \text{diag}(\beta^{i_1}, \ldots, \beta^{i_n})$ where β is the floating point base. Note

that $D^{-1}AD$ can be calculated without roundoff. When A is balanced, the computed eigenvalues are often more accurate. See Parlett and Reinsch (1969).

Problems

P7.5.1 Show that if $\bar{H} = Q^T H Q$ is obtained by performing a single-shift QR step with $H = \begin{bmatrix} w & x \\ y & z \end{bmatrix}$, then $|\bar{h}_{21}| \leq |y^2 x|/[(w-z)^2 + y^2]$.

P7.5.2 Give a formula for the 2-by-2 diagonal matrix D that minimizes $\| D^{-1}AD \|_F$ where $A = \begin{bmatrix} w & x \\ y & z \end{bmatrix}$.

P7.5.3 Explain how the single-shift QR step $H - \mu I = UR$, $\bar{H} = RU + \mu I$ can be carried out implicitly. That is, show how the transition from $\bar{H}$ to H can be carried out without subtracting the shift μ from the diagonal of H.

P7.5.4 Suppose H is upper Hessenberg and that we compute the factorization $PH = LU$ via Gaussian elimination with partial pivoting. (See Algorithm 4.3.4.) Show that $H_1 = U(P^T L)$ is upper Hessenberg and similar to H. (This is the basis of the *modified LR algorithm*.)

P7.5.5 Show that if $H = H_0$ is given and we generate the matrices H_k via $H_k - \mu_k I = U_k R_k$, $H_{k+1} = R_k U_k + \mu_k I$, then

$$(U_1 \cdots U_j)(R_j \cdots R_1) = (H - \mu_1 I) \cdots (H - \mu_j I).$$

Notes and References for Sec. 7.5

The development of the practical QR algorithm began with the important paper

H. Rutishauser (1958). "Solution of Eigenvalue Problems with the LR Transformation," *Nat. Bur. Stand. App. Math. Ser.* *49*, 47–81.

The algorithm described here was then "orthogonalized" in

J.G.F. Francis (1961). "The QR Transformation: A Unitary Analogue to the LR Transformation, Parts I and II" *Comp. J.* *4*, 265–72, 332–45.

Descriptions of the practical QR algorithm may be found in Wilkinson (1965) and Stewart (1973), and Watkins (1991). See also

D. Watkins and L. Elsner (1991). "Chasing Algorithms for the Eigenvalue Problem," *SIAM J. Matrix Anal. Appl.* *12*, 374–384.

D.S. Watkins and L. Elsner (1991). "Convergence of Algorithms of Decomposition Type for the Eigenvalue Problem," *Lin.Alg. and Its Application* *143*, 19–47.

J. Erxiong (1992). "A Note on the Double-Shift QL Algorithm," *Lin.Alg. and Its Application* *171*, 121–132.

Algol procedures for LR and QR methods are given in

R.S. Martin and J.H. Wilkinson (1968). "The Modified LR Algorithm for Complex Hessenberg Matrices," *Numer. Math.* *12*, 369–76. See also Wilkinson and Reinsch(1971, pp. 396–403).

R.S. Martin, G. Peters, and J.H. Wilkinson (1970). "The QR Algorithm for Real Hessenberg Matrices," *Numer. Math. 14*, 219–31. See also Wilkinson and Reinsch(1971, pp. 359–71).

Aspects of the balancing problem are discussed in

E.E. Osborne (1960). "On Preconditioning of Matrices," *JACM 7*, 338–45.
B.N. Parlett and C. Reinsch (1969). "Balancing a Matrix for Calculation of Eigenvalues and Eigenvectors," *Numer. Math. 13*, 292–304. See also Wilkinson and Reinsch(1971, pp. 315-26).

High performance eigenvalue solver papers include

Z. Bai and J.W. Demmel (1989). "On a Block Implementation of Hessenberg Multishift QR Iteration," *Int'l J. of High Speed Comput. 1*, 97–112.
G. Shroff (1991). "A Parallel Algorithm for the Eigenvalues and Eigenvectors of a General Complex Matrix," *Numer. Math. 58*, 779–806.
R.A. Van De Geijn (1993). "Deferred Shifting Schemes for Parallel QR Methods," *SIAM J. Matrix Anal. Appl. 14*, 180–194.
A.A. Dubrulle and G.H. Golub (1994). "A Multishift QR Iteration Without Computation of the Shifts," *Numerical Algorithms 7*, 173–181.

7.6 Invariant Subspace Computations

Several important invariant subspace problems can be solved once the real Schur decomposition $Q^TAQ = T$ has been computed. In this section we discuss how to

- compute the eigenvectors associated with some subset of $\lambda(A)$,
- compute an orthonormal basis for a given invariant subspace,
- block-diagonalize A using well-conditioned similarity transformations,
- compute a basis of eigenvectors regardless of their condition, and
- compute an approximate Jordan canonical form of A.

Eigenvector/invariant subspace computation for sparse matrices is discussed elsewhere. See §7.3 as well as portions of Chapters 8 and 9.

7.6.1 Selected Eigenvectors via Inverse Iteration

Let $q^{(0)} \in \mathbb{C}^n$ be a given unit 2-norm vector and assume that $A - \mu I \in \mathbb{R}^{n\times n}$ is nonsingular. The following is referred to as *inverse iteration*:

$$\begin{array}{l} \textbf{for } k = 1, 2, \ldots \\ \quad \text{Solve } (A - \mu I)z^{(k)} = q^{(k-1)} \\ \quad q^{(k)} = z^{(k)}/\| z^{(k)} \|_2 \\ \quad \lambda^{(k)} = q^{(k)T}Aq^{(k)} \\ \textbf{end} \end{array} \tag{7.6.1}$$

Inverse iteration is just the power method applied to $(A - \mu I)^{-1}$.

To analyze the behavior of (7.6.1), assume that A has a basis of eigenvectors $\{x_1, \ldots, x_n\}$ and that $Ax_i = \lambda_i x_i$ for $i = 1{:}n$. If

$$q^{(0)} = \sum_{i=1}^{n} \beta_i x_i$$

then $q^{(k)}$ is a unit vector in the direction of

$$(A - \mu I)^{-k} q^{(0)} = \sum_{i=1}^{n} \frac{\beta_i}{(\lambda_i - \mu)^k} x_i \, .$$

Clearly, if μ is much closer to an eigenvalue λ_j than to the other eigenvalues, then $q^{(k)}$ is rich in the direction of x_j provided $\beta_j \neq 0$.

A sample stopping criterion for (7.6.1) might be to quit as soon as the residual

$$r^{(k)} = (A - \mu I) q^{(k)}$$

satisfies

$$\| r^{(k)} \|_\infty \leq c\mathbf{u} \| A \|_\infty \tag{7.6.2}$$

where c is a constant of order unity. Since

$$(A + E_k) q^{(k)} = \mu q^{(k)}$$

with $E_k = -r^{(k)} {q^{(k)}}^T$, it follows that (7.6.2) forces μ and $q^{(k)}$ to be an exact eigenpair for a nearby matrix.

Inverse iteration can be used in conjunction with the QR algorithm as follows:

- Compute the Hessenberg decomposition $U_0^T A U_0 = H$.
- Apply the double implicit shift Francis iteration to H *without* accumulating transformations.
- For each computed eigenvalue λ whose corresponding eigenvector x is sought, apply (7.6.1) with $A = H$ and $\mu = \lambda$ to produce a vector z such that $Hz \approx \mu z$.
- Set $x = U_0 z$.

Inverse iteration with H is very economical because (1) we do not have to accumulate transformations during the double Francis iteration; (2) we can factor matrices of the form $H - \lambda I$ in $O(n^2)$ flops, and (3) only one iteration is typically required to produce an adequate approximate eigenvector.

This last point is perhaps the most interesting aspect of inverse iteration and requires some justification since λ can be comparatively inaccurate if it is ill-conditioned. Assume for simplicity that λ is real and let

$$H - \lambda I = \sum_{i=1}^{n} \sigma_i u_i v_i^T = U \Sigma V^T$$

be the SVD of $H - \lambda I$. From what we said about the roundoff properties of the QR algorithm in §7.5.6, there exists a matrix $E \in \mathbb{R}^{n \times n}$ such that $H + E - \lambda I$ is singular and $\| E \|_2 \approx \mathbf{u} \| H \|_2$. It follows that $\sigma_n \approx \mathbf{u}\sigma_1$ and $\| (H - \hat{\lambda} I) v_n \|_2 \approx \mathbf{u}\sigma_1$, i.e., v_n is a good approximate eigenvector. Clearly if the starting vector $q^{(0)}$ has the expansion

$$q^{(0)} = \sum_{i=1}^{n} \gamma_i u_i$$

then

$$z^{(1)} = \sum_{i=1}^{n} \frac{\gamma_i}{\sigma_i} v_i$$

is "rich" in the direction v_n. Note that if $s(\lambda) \approx |u_n^T v_n|$ is small, then $z^{(1)}$ is rather deficient in the direction u_n. This explains (heuristically) why another step of inverse iteration is not likely to produce an improved eigenvector approximate, especially if λ is ill-conditioned. For more details, see Peters and Wilkinson (1979).

Example 7.6.1 The matrix

$$A = \begin{bmatrix} 1 & 1 \\ 10^{-10} & 1 \end{bmatrix}$$

has eigenvalues $\lambda_1 = .99999$ and $\lambda_2 = 1.00001$ and corresponding eigenvectors $x_1 = [1,\ -10^{-5}]^T$ and $x_2 = [1,\ 10^{-5}]^T$. The condition of both eigenvalues is of order 10^5. The approximate eigenvalue $\mu = 1$ is an exact eigenvalue of $A + E$ where

$$E = \begin{bmatrix} 0 & 0 \\ -10^{-10} & 0 \end{bmatrix}.$$

Thus, the quality of μ is typical of the quality of an eigenvalue produced by the QR algorithm when executed in 10-digit floating point.

If (7.6.1) is applied with starting vector $q^{(0)} = [0,\ 1]^T$, then $q^{(1)} = [1, 0]^T$ and $\| Aq^{(1)} - \mu q^{(1)} \|_2 = 10^{-10}$. However, one more step produces $q^{(2)} = [0,\ 1]^T$ for which $\| Aq^{(2)} - \mu q^{(2)} \|_2 = 1$. This example is discussed in Peters and Wilkinson (1979).

7.6.2 Ordering Eigenvalues in the Real Schur Form

Recall that the real Schur decomposition provides information about invariant subspaces. If

$$Q^TAQ \;=\; T \;=\; \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$$
$$\qquad\qquad\qquad\qquad p \quad\; q$$

and $\lambda(T_{11}) \cap \lambda(T_{22}) = \emptyset$, then the first p columns of Q span the unique invariant subspace associated with $\lambda(T_{11})$. (See §7.1.4.) Unfortunately, the Francis iteration supplies us with a real Schur decomposition $Q_F^TAQ_F = T_F$ in which the eigenvalues appear somewhat randomly along the diagonal of T_F. This poses a problem if we want an orthonormal basis for an invariant subspace whose associated eigenvalues are not at the top of T_F's diagonal. Clearly, we need a method for computing an orthogonal matrix Q_D such that $Q_D^TT_FQ_D$ is upper quasi-triangular with appropriate eigenvalue ordering.

A look at the 2-by-2 case suggests how this can be accomplished. Suppose

$$Q_F^TAQ_F \;=\; T_F \;=\; \begin{bmatrix} \lambda_1 & t_{12} \\ 0 & \lambda_2 \end{bmatrix} \qquad \lambda_1 \neq \lambda_2$$

and that we wish to reverse the order of the eigenvalues. Note that $T_Fx = \lambda_2 x$ where

$$x \;=\; \begin{bmatrix} t_{12} \\ \lambda_2 - \lambda_1 \end{bmatrix}.$$

Let Q_D be a Givens rotation such that the second component of Q_D^Tx is zero. If $Q = Q_FQ_D$ then

$$(Q^TAQ)e_1 \;=\; Q_D^TT_F(Q_De_1) \;=\; \lambda_2Q_D^T(Q_De_1) \;=\; \lambda_2e_1$$

and so Q^TAQ must have the form

$$Q^TAQ \;=\; \begin{bmatrix} \lambda_2 & \pm t_{12} \\ 0 & \lambda_1 \end{bmatrix}.$$

By systematically interchanging adjacent pairs of eigenvalues using this technique, we can move any subset of $\lambda(A)$ to the top of T's diagonal assuming that no 2-by-2 bumps are encountered along the way.

Algorithm 7.6.1 Given an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$, an upper triangular matrix $T = Q^TAQ$, and a subset $\Delta = \{\lambda_1, \ldots, \lambda_p\}$ of $\lambda(A)$, the following algorithm computes an orthogonal matrix Q_D such that $Q_D^TTQ_D = S$ is upper triangular and $\{s_{11}, \ldots, s_{pp}\} = \Delta$. The matrices Q and T are overwritten by QQ_D and S respectively.

```
while {t_11, ..., t_pp} ≠ Δ
    for k = 1:n − 1
        if t_kk ∉ Δ and t_{k+1,k+1} ∈ Δ
            [c, s] = givens(T(k, k+1), T(k+1, k+1) − T(k, k))
            T(k:k+1, k:n) = [c s; −s c]^T T(k:k+1, k:n)
            T(1:k+1, k:k+1) = T(1:k+1, k:k+1) [c s; −s c]
            Q(1:n, k:k+1) = Q(1:n, k:k+1) [c s; −s c]
        end
    end
end
```

This algorithm requires $k(12n)$ flops, where k is the total number of required swaps. The integer k is never greater than $(n-p)p$.

The swapping gets a little more complicated when T has 2-by-2 blocks along its diagonal. See Ruhe (1970) and Stewart (1976) for details. Of course, these interchanging techniques can be used to sort the eigenvalues, say from maximum to minimum modulus.

Computing invariant subspaces by manipulating the real Schur decomposition is extremely stable. If $\hat{Q} = [\,\hat{q}_1, \ldots, \hat{q}_n\,]$ denotes the computed orthogonal matrix Q, then $\| \hat{Q}^T\hat{Q} - I \|_2 \approx \mathbf{u}$ and there exists a matrix E satisfying $\| E \|_2 \approx \mathbf{u}\| A \|_2$ such that $(A+E)\hat{q}_i \in \operatorname{span}\{\hat{q}_1, \ldots, \hat{q}_p\}$ for $i = 1{:}p$.

7.6.3 Block Diagonalization

Let

$$T = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1q} \\ 0 & T_{22} & \cdots & T_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{qq} \end{bmatrix} \begin{matrix} n_1 \\ n_2 \\ \\ n_q \end{matrix} \qquad (7.6.3)$$
$$\qquad\qquad n_1 \quad n_2 \qquad\quad n_q$$

be a partitioning of some real Schur canonical form $Q^TAQ = T \in \mathbb{R}^{n\times n}$ such that $\lambda(T_{11}), \ldots, \lambda(T_{qq})$ are disjoint. By Theorem 7.1.6 there exists a matrix Y such that $Y^{-1}TY = \operatorname{diag}(T_{11}, \ldots, T_{qq})$. A practical procedure for determining Y is now given together with an analysis of Y's sensitivity as a function of the above partitioning.

Partition $I_n = [\,E_1, \ldots, E_q\,]$ conformably with T and define the matrix

$Y_{ij} \in \mathbb{R}^{n \times n}$ as follows:

$$Y_{ij} = I_n + E_i Z_{ij} E_j^T. \qquad i < j, \; Z_{ij} \in \mathbb{R}^{n_i \times n_j}$$

In other words, Y_{ij} looks just like the identity except that Z_{ij} occupies the (i,j) block position. It follows that if $Y_{ij}^{-1} T Y_{ij} = \tilde{T} = (\tilde{T}_{ij})$ then T and $\tilde{T}$ are identical except that

$$\begin{array}{ll} \tilde{T}_{ij} = T_{ii} Z_{ij} - Z_{ij} T_{jj} + T_{ij} & \\ \tilde{T}_{ik} = T_{ik} - Z_{ij} T_{jk} & (k = j+1{:}q) \\ \tilde{T}_{kj} = T_{ki} Z_{ij} + T_{kj} & (k = 1{:}i-1) \end{array}$$

Thus, T_{ij} can be zeroed provided we have an algorithm for solving the *Sylvester equation*

$$FZ - ZG = C \tag{7.6.4}$$

where $F \in \mathbb{R}^{p \times p}$ and $G \in \mathbb{R}^{r \times r}$ are given upper quasi-triangular matrices and $C \in \mathbb{R}^{p \times r}$.

Bartels and Stewart (1972) have devised a method for doing this. Let $C = [\, c_1, \ldots, c_r \,]$ and $Z = [\, z_1, \ldots, z_r \,]$ be column partitionings. If $g_{k+1,k} = 0$, then by comparing columns in (7.6.4) we find

$$Fz_k - \sum_{i=1}^{k} g_{ik} z_i = c_k .$$

Thus, once we know $z_1, \ldots, z_{k-1}$ then we can solve the quasi-triangular system

$$(F - g_{kk} I)\, z_k = c_k + \sum_{i=1}^{k-1} g_{ik} z_i$$

for z_k. If $g_{k+1,k} \neq 0$, then z_k and z_{k+1} can be simultaneously found by solving the $2p$-by-$2p$ system

$$\begin{bmatrix} F - g_{kk} I & -g_{mk} I \\ -g_{km} I & F - g_{mm} I \end{bmatrix} \begin{bmatrix} z_k \\ z_m \end{bmatrix} = \begin{bmatrix} c_k \\ c_m \end{bmatrix} + \sum_{i=1}^{k-1} \begin{bmatrix} g_{ik} z_i \\ g_{im} z_i \end{bmatrix} \tag{7.6.5}$$

where $m = k+1$. By reordering the equations according to the permutation $(1, p+1, 2, p+2, \ldots, p, 2p)$, a banded system is obtained that can be solved in $O(p^2)$ flops. The details may be found in Bartels and Stewart (1972). Here is the overall process for the case when F and G are each triangular.

Algorithm 7.6.2 (Bartels-Stewart Algorithm) Given $C \in \mathbb{R}^{p \times r}$ and upper triangular matrices $F \in \mathbb{R}^{p \times p}$ and $G \in \mathbb{R}^{r \times r}$ that satisfy $\lambda(F) \cap \lambda(G) = \emptyset$, the following algorithm overwrites C with the solution to the equation $FZ - ZG = C$.

```
for k = 1:r
    C(1:p,k) = C(1:p,k) + C(1:p,1:k-1)G(1:k-1,k)
    Solve (F - G(k,k)I)z = C(1:p,k) for z.
    C(1:p,k) = z
end
```

This algorithm requires $pr(p+r)$ flops.

By zeroing the super diagonal blocks in T in the appropriate order, the entire matrix can be reduced to block diagonal form.

Algorithm 7.6.3 Given an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$, an upper quasi-triangular matrix $T = Q^TAQ$, and the partitioning (7.6.3), the following algorithm overwrites Q with QY where $Y^{-1}TY = \text{diag}(T_{11},\ldots,T_{qq})$.

```
for j = 2:q
    for i = 1:j-1
        Solve T_ii Z - Z T_jj = -T_ij for Z using Algorithm 7.6.2.
        for k = j+1:q
            T_ik = T_ik - Z T_jk
        end
        for k = 1:q
            Q_kj = Q_ki Z + Q_kj
        end
    end
end
```

The number of flops required by this algorithm is a complicated function of the block sizes in (7.6.3).

The choice of the real Schur form T and its partitioning in (7.6.3) determines the sensitivity of the Sylvester equations that must be solved in Algorithm 7.6.3. This in turn affects the condition of the matrix Y and the overall usefulness of the block diagonalization. The reason for these dependencies is that the relative error of the computed solution $\hat{Z}$ to

$$T_{ii}Z - ZT_{jj} \;=\; -T_{ij} \qquad (7.6.6)$$

satisfies

$$\frac{\|\hat{Z}-Z\|_F}{\|Z\|_F} \approx \mathbf{u}\frac{\|T\|_F}{\text{sep}(T_{ii},T_{jj})}.$$

For details, see Golub, Nash, and Van Loan (1979). Since

$$\text{sep}(T_{ii},T_{jj}) \;=\; \min_{X\neq 0}\frac{\|T_{ii}X - XT_{jj}\|_F}{\|X\|_F} \;\leq\; \min_{\substack{\lambda\in\lambda(T_{ii})\\ \mu\in\lambda(T_{jj})}} |\lambda - \mu|.$$

there can be a substantial loss of accuracy whenever the subsets $\lambda(T_{ii})$ are insufficiently separated. Moreover, if Z satisfies (7.6.6) then

$$\| Z \|_F \leq \frac{\| T_{ij} \|_F}{\text{sep}(T_{ii}, T_{jj})}.$$

Thus, large-norm solutions can be expected if $\text{sep}(T_{ii}, T_{jj})$ is small. This tends to make the matrix Y in Algorithm 7.6.3 ill-conditioned since it is the product of the matrices

$$Y_{ij} = \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix}.$$

Note: $\kappa_F(Y_{ij}) = 2n + \| Z \|_F^2$.

Confronted with these difficulties, Bavely and Stewart (1979) develop an algorithm for block diagonalizing that dynamically determines the eigenvalue ordering and partitioning in (7.6.3) so that all the Z matrices in Algorithm 7.6.3 are bounded in norm by some user-supplied tolerance. They find that the condition of Y can be controlled by controlling the condition of the Y_{ij}.

7.6.4 Eigenvector Bases

If the blocks in the partitioning (7.6.3) are all 1-by-1, then Algorithm 7.6.3 produces a basis of eigenvectors. As with the method of inverse iteration, the computed eigenvalue-eigenvector pairs are exact for some "nearby" matrix. A widely followed rule of thumb for deciding upon a suitable eigenvector method is to use inverse iteration whenever fewer than 25% of the eigenvectors are desired.

We point out, however, that the real Schur form can be used to determine selected eigenvectors. Suppose

$$Q^T A Q = \begin{bmatrix} T_{11} & u & T_{13} \\ 0 & \lambda & v^T \\ 0 & 0 & T_{33} \end{bmatrix} \begin{matrix} k-1 \\ 1 \\ n-k \end{matrix}$$
$$\qquad\qquad\quad k-1 \quad\; 1 \quad\; n-k$$

is upper quasi-triangular and that $\lambda \notin \lambda(T_{11}) \cup \lambda(T_{33})$. It follows that if we solve the linear systems $(T_{11} - \lambda I)w = -u$ and $(T_{33} - \lambda I)^T z = -v$ then

$$x = Q \begin{bmatrix} w \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad y = Q \begin{bmatrix} 0 \\ 1 \\ z \end{bmatrix}$$

are the associated right and left eigenvectors, respectively. Note that the condition of λ is prescribed by $1/s(\lambda) = \sqrt{(1 + w^T w)(1 + z^T z)}$.

7.6.5 Ascertaining Jordan Block Structures

Suppose that we have computed the real Schur decomposition $A = QTQ^T$, identified clusters of "equal" eigenvalues, and calculated the corresponding block diagonalization $T = Y\text{diag}(T_{11}, \ldots, T_{qq})Y^{-1}$. As we have seen, this can be a formidable task. However, even greater numerical problems confront us if we attempt to ascertain the Jordan block structure of each T_{ii}. A brief examination of these difficulties will serve to highlight the limitations of the Jordan decomposition.

Assume for clarity that $\lambda(T_{ii})$ is real. The reduction of T_{ii} to Jordan form begins by replacing it with a matrix of the form $C = \lambda I + N$, where N is the strictly upper triangular portion of T_{ii} and where λ, say, is the mean of its eigenvalues.

Recall that the dimension of a Jordan block $J(\lambda)$ is the smallest nonnegative integer k for which $[J(\lambda) - \lambda I]^k = 0$. Thus, if $p_i = \dim[\text{null}(N^i)]$, for $i = 0{:}n$, then $p_i - p_{i-1}$ equals the number of blocks in C's Jordan form that have dimension i or greater. A concrete example helps to make this assertion clear and to illustrate the role of the SVD in Jordan form computations.

Assume that C is 7-by-7. Suppose we compute the SVD $U_1^T N V_1 = \Sigma_1$ and "discover" that N has rank 3. If we order the singular values from small to large then it follows that the matrix $N_1 = V_1^T N V_1$ has the form

$$N_1 = \begin{bmatrix} 0 & K \\ 0 & L \end{bmatrix} \begin{matrix} 4 \\ 3 \end{matrix}$$
$$\qquad\quad \begin{matrix} 4 & 3 \end{matrix}$$

At this point, we know that the geometric multiplicity of λ is 4—i.e, C's Jordan form has 4 blocks ($p_1 - p_0 = 4 - 0 = 4$).

Now suppose $\tilde{U}_2^T L \tilde{V}_2 = \Sigma_2$ is the SVD of L and that we find that L has unit rank. If we again order the singular values from small to large, then $L_2 = \tilde{V}_2^T L \tilde{V}_2$ clearly has the following structure:

$$L_2 = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ 0 & 0 & c \end{bmatrix}.$$

However $\lambda(L_2) = \lambda(L) = \{0, 0, 0\}$ and so $c = 0$. Thus, if

$$V_2 = \text{diag}(I_4, \tilde{V}_2)$$

then $N_2 = V_2^T N_1 V_2$ has the following form:

$$N_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & a \\ 0 & 0 & 0 & 0 & 0 & 0 & b \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Besides allowing us to introduce more zeroes into the upper triangle, the SVD of L also enables us to deduce the dimension of the null space of N^2. Since

$$N_1^2 = \begin{bmatrix} 0 & KL \\ 0 & L^2 \end{bmatrix} = \begin{bmatrix} 0 & K \\ 0 & L \end{bmatrix} \begin{bmatrix} 0 & K \\ 0 & L \end{bmatrix}$$

and $\begin{bmatrix} K \\ L \end{bmatrix}$ has full column rank,

$$p_2 = \dim(\text{null}(N^2)) = \dim(\text{null}(N_1^2)) = 4 + \dim(\text{null}(L)) = p_1 + 2.$$

Hence, we can conclude at this stage that the Jordan form of C has at least two blocks of dimension 2 or greater.

Finally, it is easy to see that $N_1^3 = 0$, from which we conclude that there is $p_3 - p_2 = 7 - 6 = 1$ block of dimension 3 or larger. If we define $V = V_1 V_2$ then it follows that the decomposition

$$V^T C V = \begin{bmatrix} \lambda & 0 & 0 & 0 & \times & \times & \times \\ 0 & \lambda & 0 & 0 & \times & \times & \times \\ 0 & 0 & \lambda & 0 & \times & \times & \times \\ 0 & 0 & 0 & \lambda & \times & \times & \times \\ 0 & 0 & 0 & 0 & \lambda & \times & a \\ 0 & 0 & 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{4 blocks of order 1 or larger} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{2 blocks of order 2 or larger} \\ \} \text{ 1 block of order 3 or larger} \end{array}$$

"displays" C's Jordan block structure: 2 blocks of order 1, 1 block of order 2, and 1 block of order 3.

To compute the Jordan decomposition it is necessary to resort to non-orthogonal transformations. We refer the reader to either Golub and Wilkinson (1976) or Kågström and Ruhe (1980a, 1980b) for how to proceed with this phase of the reduction.

The above calculations with the SVD amply illustrate that difficult rank decisions must be made at each stage and that the final computed block structure depends critically on those decisions. Fortunately, the stable Schur decomposition can almost always be used in lieu of the Jordan decomposition in practical applications.

Problems

P7.6.1 Give a complete algorithm for solving a real, n-by-n, upper quasi-triangular system $Tx = b$.

P7.6.2 Suppose $U^{-1}AU = \text{diag}(\alpha_1, \ldots, \alpha_m)$ and $V^{-1}BV = \text{diag}(\beta_1, \ldots, \beta_n)$. Show that if $\phi(X) = AX + XB$, then $\lambda(\phi) = \{ \alpha_i + \beta_j : i = 1{:}m,\ j = 1{:}n \}$. What are the corresponding eigenvectors? How can these decompositions be used to solve $AX + XB = C$?

P7.6.3 Show that if $Y = \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix}$ then $\kappa_2(Y) = [2 + \sigma^2 + \sqrt{4\sigma^2 + \sigma^4}\ ]/2$ where $\sigma = \| Z \|_2$.

P7.6.4 Derive the system (7.6.5).

P7.6.5 Assume that $T \in \mathbb{R}^{n \times n}$ is block upper triangular and partitioned as follows:

$$T = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ 0 & T_{22} & T_{23} \\ 0 & 0 & T_{33} \end{bmatrix} \qquad T \in \mathbb{R}^{n \times n}$$

Suppose that the diagonal block T_{22} is 2-by-2 with complex eigenvalues that are disjoint from $\lambda(T_{11})$ and $\lambda(T_{33})$. Give an algorithm for computing the 2-dimensional real invariant subspace associated with T_{22}'s eigenvalues.

P7.6.6 Suppose $H \in \mathbb{R}^{n \times n}$ is upper Hessenberg with a complex eigenvalue $\lambda + i \cdot \mu$. How could inverse iteration be used to compute $x, y \in \mathbb{R}^n$ so that $H(x + iy) = \lambda + i\mu)(x + iy)$? Hint: compare real and imaginary parts in this equation and obtain a $2n$-by-$2n$ real system.

P7.6.6 (a) Prove that if $\mu_0 \in \mathbb{C}$ has nonzero real part, then the iteration

$$\mu_{k+1} = \frac{1}{2}\left(\mu_k + \frac{1}{\mu_k}\right)$$

converges to 1 if $\text{Re}(\mu_0) > 0$ and to -1 if $\text{Re}(\mu_0) < 0$. (b) Suppose $A \in \mathbb{C}^{n \times n}$ is diagonalizable and that

$$A = X \begin{bmatrix} D_+ & 0 \\ 0 & D_- \end{bmatrix} X^{-1}$$

where $D_+ \in \mathbb{C}^{p \times p}$ and $D_- \in \mathbb{C}^{(n-p) \times (n-p)}$ are diagonal with eigenvalues in the open right half plane and open left half plane, respectively. Show that the iteration

$$A_{k+1} = \frac{1}{2}\left(A_k + A_k^{-1}\right) \qquad A_0 = A$$

converges to

$$\text{sign}(A) \equiv X \begin{bmatrix} I_p & 0 \\ 0 & -I_{n-p} \end{bmatrix} X^{-1}.$$

(c) Suppose

$$M = \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix}$$
$$\qquad\quad p \qquad n-p$$

with the property that $\lambda(M_{11})$ is in the open right half plane and $\lambda(M_{22})$ is in the open left half plane. Show that

$$\text{sign}(M) = \begin{bmatrix} I_p & Z \\ 0 & -I_{n-p} \end{bmatrix}$$

and that $-Z/2$ solves $M_{11}X - XM_{22} = -M_{12}$. Thus,

$$U = \begin{bmatrix} I_p & -Z/2 \\ 0 & I_{n-p} \end{bmatrix} \Rightarrow U^{-1}MU = \begin{bmatrix} M_{11} & 0 \\ 0 & M_{22} \end{bmatrix}.$$

Notes and References for Sec. 7.6

Much of the material discussed in this section may be found in the survey paper

G.H. Golub and J.H. Wilkinson (1976). "Ill-Conditioned Eigensystems and the Computation of the Jordan Canonical Form," *SIAM Review 18*, 578–619.

Papers that specifically analyze the method of inverse iteration for computing eigenvectors include

J. Varah (1968). "The Calculation of the Eigenvectors of a General Complex Matrix by Inverse Iteration," *Math. Comp. 22*, 785–91.
J. Varah (1968). "Rigorous Machine Bounds for the Eigensystem of a General Complex Matrix," *Math. Comp. 22*, 793–801.
J. Varah (1970). "Computing Invariant Subspaces of a General Matrix When the Eigensystem is Poorly Determined," *Math. Comp. 24*, 137–49.
G. Peters and J.H. Wilkinson (1979). "Inverse Iteration, Ill-Conditioned Equations, and Newton's Method," *SIAM Review 21*, 339–60.

The Algol version of the Eispack inverse iteration subroutine is given in

G. Peters and J.H. Wilkinson (1971). "The Calculation of Specified Eigenvectors by Inverse Iteration," in Wilkinson and Reinsch (1971,pp.418–39).

The problem of ordering the eigenvalues in the real Schur form is the subject of

A. Ruhe (1970). "An Algorithm for Numerical Determination of the Structure of a General Matrix," *BIT 10*, 196–216.
G.W. Stewart (1976). "Algorithm 406: HQR3 and EXCHNG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of a Real Upper Hessenberg Matrix," *ACM Trans. Math. Soft. 2*, 275–80.
J.J. Dongarra, S. Hammarling, and J.H. Wilkinson (1992). "Numerical Considerations in Computing Invariant Subspaces," *SIAM J. Matrix Anal. Appl. 13*, 145–161.
Z. Bai and J.W. Demmel (1993). "On Swapping Diagonal Blocks in Real Schur Form," *Lin. Alg. and Its Applic. 186*, 73–95

Fortran programs for computing block diagonalizations and Jordan forms are described in

C. Bavely and G.W. Stewart (1979). "An Algorithm for Computing Reducing Subspaces by Block Diagonalization," *SIAM J. Num. Anal. 16*, 359–67.
B. Kågström and A. Ruhe (1980a). "An Algorithm for Numerical Computation of the Jordan Normal Form of a Complex Matrix," *ACM Trans. Math. Soft. 6*, 398–419.
B. Kågström and A. Ruhe (1980b). "Algorithm 560 JNF: An Algorithm for Numerical Computation of the Jordan Normal Form of a Complex Matrix," *ACM Trans. Math. Soft. 6*, 437–43.
J.W. Demmel (1983). "A Numerical Analyst's Jordan Canonical Form," Ph.D. Thesis, Berkeley.

Papers that are concerned with estimating the error in a computed eigenvalue and/or eigenvector include

S.P. Chan and B.N. Parlett (1977). "Algorithm 517: A Program for Computing the Condition Numbers of Matrix Eigenvalues Without Computing Eigenvectors," *ACM Trans. Math. Soft. 3*, 186–203.
H.J. Symm and J.H. Wilkinson (1980). "Realistic Error Bounds for a Simple Eigenvalue and Its Associated Eigenvector," *Numer. Math. 35*, 113–26.

C. Van Loan (1987). "On Estimating the Condition of Eigenvalues and Eigenvectors," *Lin. Alg. and Its Applic.* 88/89, 715–732.

Z. Bai, J. Demmel, and A. McKenney (1993). "On Computing Condition Numbers for the Nonsymmetric Eigenproblem," *ACM Trans. Math. Soft.* *19*, 202–223.

As we have seen, the sep(.,.) function is of great importance in the assessment of a computed invariant subspace. Aspects of this quantity and the associated Sylvester equation are discussed in

J. Varah (1979). "On the Separation of Two Matrices," *SIAM J. Num. Anal.* *16*, 212–22.

R. Byers (1984). "A Linpack-Style Condition Estimator for the Equation $AX - XB^T = C$," *IEEE Trans. Auto. Cont.* *AC-29*, 926–928.

K. Datta (1988). "The Matrix Equation $XA - BX = R$ and Its Applications," *Lin. Alg. and Its Appl.* *109*, 91–105.

N.J. Higham (1993). "Perturbation Theory and Backward Error for AX - XB = C," *BIT* *33*, 124–136.

J. Gardiner, M.R. Wette, A.J. Laub, J.J. Amato, and C.B. Moler (1992). "Algorithm 705: A FORTRAN-77 Software Package for Solving the Sylvester Matrix Equation $AXB^T + CXD^T = E$," *ACM Trans. Math. Soft.* *18*, 232–238.

Numerous algorithms have been proposed for the Sylvester equation, but those described in

R.H. Bartels and G.W. Stewart (1972). "Solution of the Equation $AX + XB = C$," *Comm. ACM* *15*, 820–26.

G.H. Golub, S. Nash, and C. Van Loan (1979). "A Hessenberg-Schur Method for the Matrix Problem $AX + XB = C$," *IEEE Trans. Auto. Cont.* *AC-24*, 909–13.

are among the more reliable in that they rely on orthogonal transformations. A constrained Sylvester equation problem is considerd in

J.B. Barlow, M.M. Monahemi, and D.P. O'Leary (1992). "Constrained Matrix Sylvester Equations," *SIAM J. Matrix Anal. Appl.* *13*, 1–9.

The Lyapunov problem $FX + XF^T = -C$ where C is non-negative definite has a very important role to play in control theory. See

S. Barnett and C. Storey (1968). "Some Applications of the Lyapunov Matrix Equation," *J. Inst. Math. Applic.* *4*, 33–42.

G. Hewer and C. Kenney (1988). "The Sensitivity of the Stable Lyapunov Equation," *SIAM J. Control Optim* *26*, 321–344.

A.R. Ghavimi and A.J. Laub (1995). "Residual Bounds for Discrete-Time Lyapunov Equations," *IEEE Trans. Auto. Cont.* *40*, 1244–1249.

Several authors have considered generalizations of the Sylvester equation, i.e., $\Sigma F_i X G_i = C$. These include

P. Lancaster (1970). "Explicit Solution of Linear Matrix Equations," *SIAM Review 12*, 544–66.

H. Wimmer and A.D. Ziebur (1972). "Solving the Matrix Equations $\Sigma f_p(A) g_p(A) = C$," *SIAM Review* *14*, 318–23.

W.J. Vetter (1975). "Vector Structures and Solutions of Linear Matrix Equations," *Lin. Alg. and Its Applic.* *10*, 181–88.

Some ideas about improving computed eigenvalues, eigenvectors, and invariant subspaces may be found in

J.J. Dongarra, C.B. Moler, and J.H. Wilkinson (1983). "Improving the Accuracy of Computed Eigenvalues and Eigenvectors," *SIAM J. Numer. Anal. 20,* 23–46.

J.W. Demmel (1987). "Three Methods for Refining Estimates of Invariant Subspaces," *Computing 38,* 43–57.

Hessenberg/QR iteration techniques are fast, but not very amenable to parallel computation. Because of this there is a hunger for radically new approaches to the eigenproblem. Here are some papers that focus on the matrix sign function and related ideas that have high performance potential:

C.S. Kenney and A.J. Laub (1991). "Rational Iterative Methods for the Matrix Sign Function," *SIAM J. Matrix Anal. Appl. 12,* 273–291.

C.S. Kenney, A.J. Laub, and P.M. Papadopouos (1992). "Matrix Sign Algorithms for Riccati Equations," *IMA J. of Math. Control Inform. 9,* 331–344.

C.S. Kenney and A.J. Laub (1992). "On Scaling Newton's Method for Polar Decomposition and the Matrix Sign Function," *SIAM J. Matrix Anal. Appl. 13,* 688–706.

N.J. Higham (1994). "The Matrix Sign Decomposition and Its Relation to the Polar Decomposition," *Lin. Alg. and Its Applic 212/213,* 3–20.

L. Adams and P. Arbenz (1994). "Towards a Divide and Conquer Algorithm for the Real Nonsymmetric Eigenvalue Problem," *SIAM J. Matrix Anal. Appl. 15,* 1333–1353.

7.7 The QZ Method for $\mathbf{Ax} = \lambda\mathbf{Bx}$

Let A and B be two n-by-n matrices. The set of all matrices of the form $A - \lambda B$ with $\lambda \in \mathbb{C}$ is said to be a *pencil*. The eigenvalues of the pencil are elements of the set $\lambda(A, B)$ defined by

$$\lambda(A,B) \;=\; \{z \in \mathbb{C} : \det(A - zB) = 0\,\}.$$

If $\lambda \in \lambda(A, B)$ and

$$Ax \;=\; \lambda Bx \qquad x \neq 0 \tag{7.7.1}$$

then x is referred to as an eigenvector of $A - \lambda B$.

In this section we briefly survey some of the mathematical properties of the generalized eigenproblem (7.7.1) and present a stable method for its solution. The important case when A and B are symmetric with the latter positive definite is discussed in §8.7.2.

7.7.1 Background

The first thing to observe about the generalized eigenvalue problem is that there are n eigenvalues if and only if $\text{rank}(B) = n$. If B is rank deficient

then $\lambda(A, B)$ may be finite, empty, or infinite:

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \Rightarrow \quad \lambda(A,B) = \{1\}$$

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \Rightarrow \quad \lambda(A,B) = \emptyset$$

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \Rightarrow \quad \lambda(A,B) = \mathbb{C}$$

Note that if $0 \neq \lambda \in \lambda(A, B)$ then $(1/\lambda) \in \lambda(B, A)$. Moreover, if B is nonsingular then $\lambda(A, B) = \lambda(B^{-1}A, I) = \lambda(B^{-1}A)$.

This last observation suggests one method for solving the $A - \lambda B$ problem when B is nonsingular:

- Solve $BC = A$ for C using (say) Gaussian elimination with pivoting.
- Use the QR algorithm to compute the eigenvalues of C.

Note that C will be affected by roundoff errors of order $\mathbf{u}\| A \|_2 \| B^{-1} \|_2$. If B is ill-conditioned, then this can rule out the possibility of computing any generalized eigenvalue accurately—even those eigenvalues that may be regarded as well-conditioned.

Example 7.7.1 If

$$A = \begin{bmatrix} 1.746 & .940 \\ 1.246 & 1.898 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} .780 & .563 \\ .913 & .659 \end{bmatrix}$$

then $\lambda(A, B) = \{2, 1.07 \times 10^6\}$. With 7-digit floating point arithmetic, we find $\lambda(fl(AB^{-1}))$ $= \{1.562539,\ 1.01 \times 10^6\}$. The poor quality of the small eigenvalue is because $\kappa_2(B) \approx 2 \times 10^6$. On the other hand, we find that

$$\lambda(I, fl(A^{-1}B)) \approx \{2.000001,\ 1.06 \times 10^6\}.$$

The accuracy of the small eigenvalue is improved because $\kappa_2(A) \approx 4$.

Example 7.7.1 suggests that we seek an alternative approach to the $A - \lambda B$ problem. One idea is to compute well-conditioned Q and Z such that the matrices

$$A_1 = Q^{-1}AZ \qquad B_1 = Q^{-1}BZ \tag{7.7.2}$$

are each in canonical form. Note that $\lambda(A, B) = \lambda(A_1, B_1)$ since

$$Ax = \lambda Bx \quad \Leftrightarrow \quad A_1 y = \lambda B_1 y \qquad x = Zy$$

We say that the pencils $A - \lambda B$ and $A_1 - \lambda B_1$ are *equivalent* if (7.7.2) holds with nonsingular Q and Z.

7.7.2 The Generalized Schur Decomposition

As in the standard eigenproblem $A - \lambda I$ there is a choice between canonical forms. Analogous to the Jordan form is a decomposition of Kronecker in which both A_1 and B_1 are block diagonal. The blocks are similar to Jordan blocks. The Kronecker canonical form poses the same numerical difficulties as the Jordan form. However, this decomposition does provide insight into the mathematical properties of the pencil $A - \lambda B$. See Wilkinson (1978) and Demmel and Kågström (1987) for details.

More attractive from the numerical point of view is the following decomposition described in Moler and Stewart (1973).

Theorem 7.7.1 (Generalized Schur Decomposition) *If A and B are in $\mathbb{C}^{n\times n}$, then there exist unitary Q and Z such that $Q^H AZ = T$ and $Q^H BZ = S$ are upper triangular. If for some k, t_{kk} and s_{kk} are both zero, then $\lambda(A, B) = \mathbb{C}$. Otherwise*

$$\lambda(A, B) = \{t_{ii}/s_{ii} : s_{ii} \neq 0\}.$$

Proof. Let $\{B_k\}$ be a sequence of nonsingular matrices that converge to B. For each k, let $Q_k^H(AB_k^{-1})Q_k = R_k$ be a Schur decomposition of AB_k^{-1}. Let Z_k be unitary such that $Z_k^H(B_k^{-1}Q_k) \equiv S_k^{-1}$ is upper triangular. It follows that both $Q_k^H AZ_k = R_k S_k$ and $Q_k^H B_k Z_k = S_k$ are also upper triangular.

Using the Bolzano-Weierstrass theorem, we know that the bounded sequence $\{(Q_k, Z_k)\}$ has a converging subsequence, $\lim(Q_{k_i}, Z_{k_i}) = (Q, Z)$. It is easy to show that Q and Z are unitary and that $Q^H AZ$ and $Q^H BZ$ are upper triangular. The assertions about $\lambda(A, B)$ follow from the identity

$$\det(A - \lambda B) = \det(QZ^H)\prod_{i=1}^{n}(t_{ii} - \lambda s_{ii}). \ \square$$

If A and B are real then the following decomposition, which corresponds to the real schur decomposition (Theorem 7.4.1), is of interest.

Theorem 7.7.2 (Generalized Real Schur Decomposition) *If A and B are in $\mathbb{R}^{n\times n}$ then there exist orthogonal matrices Q and Z such that $Q^T AZ$ is upper quasi-triangular and $Q^T BZ$ is upper triangular.*

Proof. See Stewart (1972). $\square$

In the remainder of this section we are concerned with the computation of this decomposition and the mathematical insight that it provides.

7.7.3 Sensitivity Issues

The generalized Schur decomposition sheds light on the issue of eigenvalue sensitivity for the $A - \lambda B$ problem. Clearly, small changes in A and B can

induce large changes in the eigenvalue $\lambda_i = t_{ii}/s_{ii}$ if s_{ii} is small. However, as Stewart (1978) argues, it may not be appropriate to regard such an eigenvalue as "ill-conditioned." The reason is that the reciprocal $\mu_i = s_{ii}/t_{ii}$ might be a very well behaved eigenvalue for the pencil $\mu A - B$. In the Stewart analysis, A and B are treated symmetrically and the eigenvalues are regarded more as ordered pairs (t_{ii}, s_{ii}) than as quotients. With this point of view it becomes appropriate to measure eigenvalue perturbations in the *chordal metric* chord (a, b) defined by

$$\text{chord}(a,b) \;=\; \frac{|a-b|}{\sqrt{1+a^2}\sqrt{1+b^2}}\,.$$

Stewart shows that if λ is a distinct eigenvalue of $A - \lambda B$ and λ_ϵ is the corresponding eigenvalue of the perturbed pencil $\tilde{A} - \lambda\tilde{B}$ with $\| A - \tilde{A} \|_2 \approx \| B - \tilde{B} \|_2 \approx \epsilon$, then

$$\text{chord}(\lambda, \lambda_\epsilon) \;\le\; \frac{\epsilon}{(y^HAx)^2 + (y^HBx)^2} \;+\; O(\epsilon^2)$$

where x and y have unit 2-norm and satisfy $Ax = \lambda Bx$ and $y^H = \lambda y^H B$. Note that the denominator in the upper bound is symmetric in A and B. The "truly" ill-conditioned eigenvalues are those for which this denominator is small.

The extreme case when $t_{kk} = s_{kk} = 0$ for some k has been studied by Wilkinson (1979). He makes the interesting observation that when this occurs, the remaining quotients t_{ii}/s_{ii} can assume arbitrary values.

7.7.4 Hessenberg-Triangular Form

The first step in computing the generalized Schur decomposition of the pair (A, B) is to reduce A to upper Hessenberg form and B to upper triangular form via orthogonal transformations. We first determine an orthogonal U such that U^TB is upper triangular. Of course, to preserve eigenvalues, we must also update A in exactly the same way. Let's trace what happens in the $n = 5$ case.

$$A = U^TA = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}, B = U^TB = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

Next, we reduce A to upper Hessenberg form while preserving B's upper triangular form. First, a Givens rotation Q_{45} is determined to zero a_{51}:

$$A = Q_{45}^T A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}, B = Q_{45}^T B = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

The nonzero entry arising in the (5,4) position in B can be zeroed by postmultiplying with an appropriate Givens rotation Z_{45} :

$$A = AZ_{45} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}, B = BZ_{45} = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

Zeros are similarly introduced into the (4, 1) and (3, 1) positions in A:

$$A = Q_{34}^T A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}, B = Q_{34}^T B = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$A = AZ_{34} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}, B = BZ_{34} = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$A = Q_{23}^T A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}, B = Q_{23}^T B = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$A = AZ_{23} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}, B = BZ_{23} = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

A is now upper Hessenberg through its first column. The reduction is completed by zeroing a_{52}, a_{42}, and a_{53}. As is evident above, two orthogonal transformations are required for each a_{ij} that is zeroed—one to do the

zeroing and the other to restore B's triangularity. Either Givens rotations or 2-by-2 modified Householder transformations can be used. Overall we have:

Algorithm 7.7.1 (Hessenberg-Triangular Reduction) Given A and B in $\mathbb{R}^{n\times n}$, the following algorithm overwrites A with an upper Hessenberg matrix Q^TAZ and B with an upper triangular matrix Q^TBZ where both Q and Z are orthogonal.

Using Algorithm 5.2.1, overwrite B with $Q^TB = R$ where
  Q is orthogonal and R is upper triangular.
$A = Q^TA$
for $j = 1{:}n-2$
  for $i = n{:}-1{:}j+2$
    $[c,s] = \textbf{givens}(A(i-1,j), A(i,j))$
    $A(i-1{:}i, j{:}n) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T A(i-1{:}i, j{:}n)$
    $B(i-1{:}i, i-1{:}n) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T B(i-1{:}i, i-1{:}n)$
    $[c,s] = \textbf{givens}(-B(i,i), B(i,i-1))$
    $B(1{:}i, i-1{:}i) = B(1{:}i, i-1{:}i) \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$
    $A(1{:}n, i-1{:}i) = A(1{:}n, i-1{:}i) \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$
  end
end

This algorithm requires about $8n^3$ flops. The accumulation of Q and Z requires about $4n^3$ and $3n^3$ flops, respectively.

The reduction of $A - \lambda B$ to Hessenberg-triangular form serves as a "front end" decomposition for a generalized QR iteration known as the QZ iteration which we describe next.

Example 7.7.3 If

$$A = \begin{bmatrix} 10 & 1 & 2 \\ 1 & 2 & -1 \\ 1 & 1 & 2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

and orthogonal matrices Q and Z are defined by

$$Q = \begin{bmatrix} -.1231 & -.9917 & .0378 \\ -.4924 & .0279 & -.8699 \\ -.8616 & .1257 & .4917 \end{bmatrix} \quad \text{and} \quad Z = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & -.8944 & -.4472 \\ 0.0000 & .4472 & -.8944 \end{bmatrix}$$

then $A_1 = Q^TAZ$ and $B_1 = Q^TBZ$ are given by

$$A_1 = \begin{bmatrix} -2.5849 & 1.5413 & 2.4221 \\ -9.7631 & .0874 & 1.9239 \\ 0.0000 & 2.7233 & -.7612 \end{bmatrix} \quad \text{and} \quad B_1 = \begin{bmatrix} -8.1240 & 3.6332 & 14.2024 \\ 0.0000 & 0.0000 & 1.8739 \\ 0.0000 & 0.0000 & .7612 \end{bmatrix}.$$

7.7.5 Deflation

In describing the QZ iteration we may assume without loss of generality that A is an unreduced upper Hessenberg matrix and that B is a nonsingular upper triangular matrix. The first of these assertions is obvious, for if $a_{k+1,k} = 0$ then

$$A - \lambda B = \begin{bmatrix} A_{11} - \lambda B_{11} & A_{12} - \lambda B_{12} \\ 0 & A_{22} - \lambda B_{22} \end{bmatrix} \begin{matrix} k \\ n-k \end{matrix}$$
$$\qquad\qquad\qquad k \qquad\qquad n-k$$

and we may proceed to solve the two smaller problems $A_{11} - \lambda B_{11}$ and $A_{22} - \lambda B_{22}$. On the other hand, if $b_{kk} = 0$ for some k, then it is possible to introduce a zero in A's $(n, n-1)$ position and thereby deflate. Illustrating by example, suppose $n = 5$ and $k = 3$:

$$A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}, \quad B = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

The zero on B's diagonal can be "pushed down" to the (5,5) position as follows using Givens rotations:

$$A = Q_{34}^T A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}, B = Q_{34}^T B = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$A = AZ_{23} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}, B = BZ_{23} = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$A = Q_{45}^T A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}, B = Q_{45}^T B = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A = AZ_{34} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}, B = BZ_{34}^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A = AZ_{45} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}, B = BZ_{45} = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This zero-chasing technique is perfectly general and can be used to zero $a_{n,n-1}$ regardless of where the zero appears along B's diagonal.

7.7.6 The QZ Step

We are now in a position to describe a QZ step. The basic idea is to update A and B as follows

$$(\bar{A} - \lambda \bar{B}) = \bar{Q}^T (A - \lambda B) \bar{Z},$$

where $\bar{A}$ is upper Hessenberg, $\bar{B}$ is upper triangular, $\bar{Q}$ and $\bar{Z}$ are each orthogonal, and $\bar{A}\bar{B}^{-1}$ is essentially the same matrix that would result if a Francis QR step (Algorithm 7.5.2) were explicitly applied to AB^{-1}. This can be done with some clever zero-chasing and an appeal to the implicit Q theorem.

Let $M = AB^{-1}$ (upper Hessenberg) and let v be the first column of the matrix $(M - aI)(M - bI)$, where a and b are the eigenvalues of M's lower 2-by-2 submatrix. Note that v can be calculated in $O(1)$ flops. If P_0 is a Householder matrix such that $P_0 v$ is a multiple of e_1, then

$$A = P_0 A = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

$$B = P_0 B = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}.$$

The idea now is to restore these matrices to Hessenberg-triangular form by chasing the unwanted nonzero elements down the diagonal.

To this end, we first determine a pair of Householder matrices Z_1 and

Z_2 to zero b_{31}, b_{32}, and b_{21}:

$$A = AZ_1Z_2 = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

$$B = BZ_1Z_2 = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

Then a Householder matrix P_1 is used to zero a_{31} and a_{41}:

$$A = P_1A = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

$$B = P_1B = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

Notice that with this step the unwanted nonzero elements have been shifted down and to the right from their original position. This illustrates a typical step in the QZ iteration. Notice that $Q = Q_0Q_1 \cdots Q_{n-2}$ has the same first column as Q_0. By the way the initial Householder matrix was determined, we can apply the implicit Q theorem and assert that $AB^{-1} = Q^T(AB^{-1})Q$ is indeed essentially the same matrix that we would obtain by applying the Francis iteration to $M = AB^{-1}$ directly. Overall we have:

Algorithm 7.7.2 (The QZ Step) Given an unreduced upper Hessenberg matrix $A \in \mathbb{R}^{n\times n}$ and a nonsingular upper triangular matrix $B \in \mathbb{R}^{n\times n}$, the following algorithm overwrites A with the upper Hessenberg matrix Q^TAZ and B with the upper triangular matrix Q^TBZ where Q and Z are orthogonal and Q has the same first column as the orthogonal similarity transformation in Algorithm 7.5.1 when it is applied to AB^{-1}.

Let $M = AB^{-1}$ and compute $(M - aI)(M - bI)e_1 = (x, y, z, 0, \ldots, 0)^T$
    where a and b are the eigenvalues of M's lower 2-by-2.
for $k = 1{:}n - 2$
    Find Householder Q_k so $Q_k[\,x\; y\; z\,]^T = [\,*\; 0\; 0\,]^T$.
    $A = \text{diag}(I_{k-1}, Q_k, I_{n-k-2})A$
    $B = \text{diag}(I_{k-1}, Q_k, I_{n-k-2})B$
    Find Householder Z_{k1} so
        $\begin{bmatrix} b_{k+2,k} & b_{k+2,k+1} & b_{k+2,k+2} \end{bmatrix} Z_{k1} = \begin{bmatrix} 0 & 0 & * \end{bmatrix}$.
    $A = A\text{diag}(I_{k-1}, Z_{k1}, I_{n-k-2})$
    $B = B\text{diag}(I_{k-1}, Z_{k1}, I_{n-k-2})$
    Find Householder Z_{k2} so
        $\begin{bmatrix} b_{k+1,k} & b_{k+1,k+1} \end{bmatrix} Z_{k2} = \begin{bmatrix} 0 & * \end{bmatrix}$.
    $A = A\text{diag}(I_{k-1}, Z_{k2}, I_{n-k-1})$
    $B = B\text{diag}(I_{k-1}, Z_{k2}, I_{n-k-1})$
    $x = a_{k+1,k};\; y = a_{k+1,k}$
    if $k < n - 2$
        $z = a_{k+3,k}$
    end
end
Find Householder Q_{n-1} so $Q_{n-1} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix}$
$A = \text{diag}(I_{n-2}, Q_{n-1})A$
$B = \text{diag}(I_{n-2}, Q_{n-1})B$
Find Householder Z_{n-1} so
    $\begin{bmatrix} b_{n,n-1} & b_{nn} \end{bmatrix} Z_{n-1} = \begin{bmatrix} 0 & * \end{bmatrix}$
$A = A\text{diag}(I_{n-2}, Z_{n-1})$
$B = B\text{diag}(I_{n-2}, Z_{n-1})$

This algorithm requires $22n^2$ flops. Q and Z can be accumulated for an additional $8n^2$ flops and $13n^2$ flops, respectively.

7.7.7 The Overall QZ Process

By applying a sequence of QZ steps to the Hessenberg-triangular pencil $A - \lambda B$, it is possible to reduce A to quasi-triangular form. In doing this it is necessary to monitor A's subdiagonal and B's diagonal in order to bring about decoupling whenever possible. The complete process, due to Moler and Stewart (1973), is as follows:

Algorithm 7.7.3 Given $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$, the following algorithm computes orthogonal Q and Z such that $Q^T A Z = T$ is upper quasi-triangular and $Q^T B Z = S$ is upper triangular. A is overwritten by T and B by S.

Using Algorithm 7.7.1, overwrite A with $Q^T AZ$ (upper Hessenberg) and B with $Q^T BZ$ (upper triangular).

until $q = n$

Set all subdiagonal elements in A to zero that satisfy $|a_{i,i-1}| \leq \epsilon(|a_{i-1,i-1}| + |a_{ii}|)$

Find the largest nonnegative q and the smallest nonnegative p such that if

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & 0 & A_{33} \end{bmatrix} \begin{matrix} p \\ n \quad p \quad q \\ q \end{matrix}$$

$$\qquad p \qquad n-p-q \qquad q$$

then A_{33} is upper quasi-triangular and A_{22} is unreduced upper Hessenberg.

Partition B conformably:

$$B - \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ 0 & B_{22} & B_{23} \\ 0 & 0 & B_{33} \end{bmatrix} \begin{matrix} p \\ n-p-q \\ q \end{matrix}$$

$$\qquad p \qquad n-p-q \qquad q$$

if $q < n$

if B_{22} is singular

Zero $a_{n-q,n-q \cdot 1}$

else

Apply Algorithm 7.7.2 to A_{22} and B_{22}

$A = \text{diag}(I_p, Q, I_q)^T A\text{diag}(I_p, Z, I_q)$

$B = \text{diag}(I_p, Q, I_q)^T B\text{diag}(I_p, Z, I_q)$

end

end

end

This algorithm requires $30n^3$ flops. If Q is desired, an additional $16n^3$ are necessary. If Z is required, an additional $20n^3$ are needed. These estimates of work are based on the experience that about two QZ iterations per eigenvalue are necessary. Thus, the convergence properties of QZ are the same as for QR. The speed of the QZ algorithm is not affected by rank deficiency in B.

The computed S and T can be shown to satisfy

$$Q_0^T(A+E)Z_0 = T \qquad Q_0^T(B+F)Z_0 = S$$

where Q_0 and Z_0 are exactly orthogonal and $\| E \|_2 \approx \mathbf{u}\| A \|_2$ and $\| F \|_2 \approx \mathbf{u}\| B \|_2$.

Example 7.7.5 If the QZ algorithm is applied to

$$A = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 4 & 4 & 5 & 6 & 7 \\ 0 & 3 & 6 & 7 & 8 \\ 0 & 0 & 2 & 8 & 9 \\ 0 & 0 & 0 & 1 & 10 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & -1 & -1 & -1 & -1 \\ 0 & 1 & -1 & -1 & -1 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

then the subdiagonal elements of A converge as follows

Iteration	$O(\|h_{21}\|)$	$O(\|h_{32}\|)$	$O(\|h_{43}\|)$	$O(\|h_{54}\|)$
1	10_0	10^1	10^0	10^{-1}
2	10^0	10^0	10^0	10^{-1}
3	10^0	10^1	10^{-1}	10^{-3}
4	10^0	10^0	10^{-1}	10^{-8}
5	10^0	10^1	10^{-1}	10^{-16}
6	10^0	10^0	10^{-2}	converg.
7	10^0	10^{-1}	10^{-4}	
8	10^1	10^{-1}	10^{-8}	
9	10^0	10^{-1}	10^{-19}	
10	10^0	10^{-2}	converg.	
11	10^{-1}	10^{-4}		
12	10^{-2}	10^{-11}		
13	10^{-3}	10^{-27}		
14	converg.	converg.		

7.7.8 Generalized Invariant Subspace Computations

Many of the invariant subspace computations discussed in §7.6 carry over to the generalized eigenvalue problem. For example, approximate eigenvectors can be found via inverse iteration:

$q^{(0)} \in \mathbb{C}^{n\times n}$ given.
for $k = 1, 2, \ldots$
  Solve $(A - \mu B)z^{(k)} = Bq^{(k-1)}$
  Normalize: $q^{(k)} = z^{(k)}/\| z^{(k)} \|_2$
  $\lambda^{(k)} = [q^{(k)}]^H A q^{(k)} \;/\; [q^{(k)}]^H A q^{(k)}$
end

When B is nonsingular, this is equivalent to applying (7.6.1) with the matrix $B^{-1}A$. Typically, only a single iteration is required if μ is an approximate eigenvalue computed by the QZ algorithm. By inverse iterating with the Hessenberg-triangular pencil, costly accumulation of the Z-transformations during the QZ iteration can be avoided.

Corresponding to the notion of an invariant subspace for a single matrix, we have the notion of a *deflating* subspace for the pencil $A - \lambda B$. In

particular, we say that a k-dimensional subspace $S \subseteq \mathbb{R}^n$ is "deflating" for the pencil $A - \lambda B$ if the subspace $\{ Ax + By : x, y \in S \}$ has dimension k or less. Note that the columns of the matrix Z in the generalized Schur decomposition define a family of deflating subspaces, for if $Q = [\, q_1, \ldots, q_n \,]$ and $Z = [\, z_1, \ldots, z_n \,]$ then we have $\text{span}\{Az_1, \ldots, Az_k\} \subseteq \text{span}\{q_1, \ldots, q_k\}$ and $\text{span}\{Bz_1, \ldots, Bz_k\} \subseteq \text{span}\{q_1, \ldots, q_k\}$. Properties of deflating subspaces and their behavior under perturbation are described in Stewart (1972).

Problems

P7.7.1 Suppose A and B are in $\mathbb{R}^{n \times n}$ and that

$$U^T B V = \underset{\quad\; r \quad\; n-r}{\left[\begin{array}{cc} D & 0 \\ 0 & 0 \end{array} \right]} \begin{array}{l} r \\ n-r \end{array} \qquad U = \underset{\qquad r \qquad\; n-r}{[\; U_1 \quad U_2 \;]} \qquad V = \underset{\qquad r \qquad\; n-r}{[\; V_1 \quad V_2 \;]}$$

is the SVD of B, where D is r-by-r and $r = \text{rank}(B)$. Show that if $\lambda(A, B) = \mathbb{C}$ then $U_2^T A V_2$ is singular.

P7.7.2 Define $\text{F} : \mathbb{R}^n \to \mathbb{R}$ by

$$\text{F}(x) \;=\; \frac{1}{2} \left\| Ax \;-\; \frac{x^T B^T A x}{x^T B^T B x}\, Bx \right\|_2^2$$

where A and B are in $\mathbb{R}^{m \times n}$. Show that if $\nabla \text{F}(x) = 0$, then Ax is a multiple of Bx.

P7.7.3 Suppose A and B are in $\mathbb{R}^{n \times n}$. Give an algorithm for computing orthogonal Q and Z such that $Q^T A Z$ is upper Hessenberg and $Z^T B Q$ is upper triangular.

P7.7.4 Suppose

$$A = \left[\begin{array}{cc} A_{11} & A_{12} \\ 0 & A_{22} \end{array} \right] \quad \text{and} \quad B = \left[\begin{array}{cc} B_{11} & B_{12} \\ 0 & B_{22} \end{array} \right]$$

with $A_{11}, B_{11} \in \mathbb{R}^{k \times k}$ and $A_{22}, B_{22} \in \mathbb{R}^{j \times j}$. Under what circumstances do there exist

$$X = \left[\begin{array}{cc} I_k & X_{12} \\ 0 & I_j \end{array} \right] \quad \text{and} \quad Y = \left[\begin{array}{cc} I_k & Y_{12} \\ 0 & I_j \end{array} \right]$$

so that $Y^{-1}AX$ and $Y^{-1}BX$ are both block diagonal? This is the *generalized Sylvester equation problem.* Specify an algorithm for the case when A_{11}, A_{22}, B_{11}, and B_{22} are upper triangular. See Kågström (1994).

P7.7.5 Suppose $\mu \notin \lambda(A, B)$. Relate the eigenvalues and eigenvectors of $A_1 = (A - \mu B)^{-1} A$ and $B_1 = (A - \mu B)^{-1} B$ to the generalized eigenvalues and eigenvectors of $A - \lambda B$.

P7.7.6 Suppose $A, B, C, D \in \mathbb{R}^{n \times n}$. Show how to compute orthogonal matrices Q, Z, U, and V such that $Q^T A U$ is upper Hessenberg and $V^T C Z$, $Q^T B V$, and $V^T D Z$ are all upper triangular. Note that this converts the pencil $AC - \lambda BD$ to Hessenberg-triangular form. Your algorithm should not form the products AC or BD explicitly and not should not compute any matrix inverse. See Van Loan (1975).

Notes and References for Sec. 7.7

Mathematical aspects of the generalized eigenvalue problem are covered in

F. Gantmacher (1959). *The Theory of Matrices* , vol. 2, Chelsea, New York.
H.W. Turnbill and A.C. Aitken (1961). *An Introduction to the Theory of Canonical Matrices*, Dover, New York.
I. Erdelyi (1967). "On the Matrix Equation $Ax = \lambda Bx$," *J. Math. Anal. and Applic.* *17*, 119–32.

A good general volume that covers many aspects of the $A - \lambda B$ problem is

B. Kågström and A. Ruhe (1983). *Matrix Pencils*, Proc. Pite Havsbad, 1982, Lecture Notes in Mathematics 973, Springer-Verlag, New York and Berlin.

The perturbation theory for the generalized eigenvalue problem is treated in

G.W. Stewart (1972). "On the Sensitivity of the Eigenvalue Problem $Ax = \lambda Bx$," *SIAM J. Num. Anal. 9*, 669–86.
G.W. Stewart (1973). "Error and Perturbation Bounds for Subspaces Associated with Certain Eigenvalue Problems," *SIAM Review 15*, 727–64.
G.W. Stewart (1975). "Gershgorin Theory for the Generalized Eigenvalue Problem $Ax = \lambda Bx$," *Math. Comp. 29*, 600–606.
G.W. Stewart (1978). "Perturbation Theory for the Generalized Eigenvalue Problem'", in *Recent Advances in Numerical Analysis* , ed. C. de Boor and G.H. Golub, Academic Press, New York.
A. Pokrzywa (1986). "On Perturbations and the Equivalence Orbit of a Matrix Pencil," *Lin. Alg. and Applic. 82*, 99–121.

The QZ-related papers include

C.B. Moler and G.W. Stewart (1973). "An Algorithm for Generalized Matrix Eigenvalue Problems," *SIAM J. Num. Anal. 10*, 241–56.
L. Kaufman (1974). "The LZ Algorithm to Solve the Generalized Eigenvalue Problem," *SIAM J. Num. Anal. 11*, 997–1024.
R.C. Ward (1975). "The Combination Shift QZ Algorithm," *SIAM J. Num. Anal. 12*, 835–853.
C.F. Van Loan (1975). "A General Matrix Eigenvalue Algorithm," *SIAM J. Num. Anal. 12*, 819–834.
L. Kaufman (1977). "Some Thoughts on the QZ Algorithm for Solving the Generalized Eigenvalue Problem," *ACM Trans. Math. Soft. 3*, 65–75.
R.C. Ward (1981). "Balancing the Generalized Eigenvalue Problem," *SIAM J. Sci and Stat. Comp. 2*, 141–152.
P. Van Dooren (1982). "Algorithm 590: DSUBSP and EXCHQZ: Fortran Routines for Computing Deflating Subspaces with Specified Spectrum," *ACM Trans. Math. Software 8*, 376–382.
D. Watkins and L. Elsner (1994). "Theory of Decomposition and Bulge-Chasing Algorithms for the Generalized Eigenvalue Problem," *SIAM J. Matrix Anal. Appl. 15*, 943–967.

Just as the Hessenberg decomposition is important in its own right, so is the Hessenberg-triangular decomposition that serves as a QZ front end. See

W. Enright and S. Serbin (1978). "A Note on the Efficient Solution of Matrix Pencil Systems," *BIT 18*, 276–81.

Other solution frameworks are proposed in

V.N. Kublanovskaja and V.N. Fadeeva (1964). "Computational Methods for the Solution of a Generalized Eigenvalue Problem," *Amer. Math. Soc. Transl. 2*, 271–90.
G. Peters and J.H. Wilkinson (1970a). "$Ax = \lambda Bx$ and the Generalized Eigenproblem," *SIAM J. Num. Anal. 7*, 479–92.

G. Rodrigue (1973). "A Gradient Method for the Matrix Eigenvalue Problem $Ax = \lambda Bx$," *Numer. Math.* *22,* 1–16.
H.R. Schwartz (1974). "The Method of Coordinate Relaxation for $(A - \lambda B)x = 0$," *Num. Math.* *23,* 135–52.
A. Jennings and M.R. Osborne (1977). "Generalized Eigenvalue Problems for Certain Unsymmetric Band Matrices," *Lin. Alg. and Its Applic.* *29,* 139–50.
V.N. Kublanovskaya (1984). "AB Algorithm and Its Modifications for the Spectral Problem of Linear Pencils of Matrices," *Numer. Math.* *43,* 329–342.
C. Oara (1994). "Proper Deflating Subspaces: Properties, Algorithms, and Applications," *Numerical Algorithms* *7,* 355–373.

The general $Ax = \lambda Bx$ problem is central to some important control theory applications. See

P. Van Dooren (1981). "A Generalized Eigenvalue Approach for Solving Riccati Equations," *SIAM J. Sci. and Stat. Comp.* *2,* 121–135.
P. Van Dooren (1981). "The Generalized Eigenstructure Problem in Linear System Theory," *IEEE Trans. Auto. Cont. AC-26,* 111–128.
W.F. Arnold and A.J. Laub (1984). "Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations," *Proc. IEEE* *72,* 1746–1754.
J.W. Demmel and B. Kågström (1988). "Accurate Solutions of Ill-Posed Problems in Control Theory," *SIAM J. Matrix Anal. Appl.* 126–145.
U. Flaschka, W-W. Li, and J-L. Wu (1992). "A KQZ Algorithm for Solving Linear-Response Eigenvalue Equations," *Lin. Alg. and Its Applic.* *165,* 93–123.

Rectangular generalized eigenvalue problems arise in certain applications. See

G.L. Thompson and R.L. Weil (1970). "Reducing the Rank of $A - \lambda B$," *Proc. Amer. Math. Sec.* *26,* 548–54.
G.L. Thompson and R.L. Weil (1972). "Roots of Matrix Pencils $Ay = \lambda By$: Existence, Calculations, and Relations to Game Theory," *Lin. Alg. and Its Applic.* *5,* 207–26.
G.W. Stewart (1994). "Perturbation Theory for Rectangular Matrix Pencils," *Lin. Alg. and Applic.* *208/209,* 297–301.

The *Kronecker Structure* of the pencil $A - \lambda B$ is analogous to Jordan structure of $A - \lambda I$: it provides very useful information about the underlying application.

J.H. Wilkinson (1978). "Linear Differential Equations and Kronecker's Canonical Form," in *Recent Advances in Numerical Analysis* , ed. C. de Boor and G.H. Golub, Academic Press, New York, pp. 231–65.

Interest in the Kronecker structure has led to a host of new algorithms and analyses.

J.H. Wilkinson (1979). "Kronecker's Canonical Form and the QZ Algorithm," *Lin. Alg. and Its Applic.* *28,* 285–303.
P. Van Dooren (1979). "The Computation of Kronecker's Canonical Form of a Singular Pencil," *Lin. Alg. and Its Applic.* *27,* 103–40.
J.W. Demmel (1983). "The Condition Number of Equivalence Transformations that Block Diagonalize Matrix Pencils," *SIAM J. Numer. Anal.* *20,* 599–610.
J.W. Demmel and B. Kågström (1987). "Computing Stable Eigendecompositions of Matrix Pencils," *Linear Alg. and Its Applic* *88/89,* 139–186.
B. Kågström (1985). "The Generalized Singular Value Decomposition and the General $A - \lambda B$ Problem," *BIT* *24,* 568–583.
B. Kågström (1986). "RGSVD: An Algorithm for Computing the Kronecker Structure and Reducing Subspaces of Singular $A - \lambda B$ Pencils," *SIAM J. Sci. and Stat. Comp.* *7,*185–211.
J. Demmel and B. Kågström (1986). "Stably Computing the Kronecker Structure and Reducing Subspaces of Singular Pencils $A - \lambda B$ for Uncertain Data," in *Large Scale*

Eigenvalue Problems, J. Cullum and R.A. Willoughby (eds), North-Holland, Amsterdam.

T. Beelen and P. Van Dooren (1988). "An Improved Algorithm for the Computation of Kronecker's Canonical Form of a Singular Pencil," *Lin. Alg. and Its Applic.* *105*, 9–65.

B. Kågström and L. Westin (1989). "Generalized Schur Methods with Condition Estimators for Solving the Generalized Sylvester Equation," *IEEE Trans. Auto. Cont. AC-34*, 745–751.

B. Kågström and P. Poromaa (1992). "Distributed and Shared Memory Block Algorithms for the Triangular Sylvester Equation with sep^{-1} Estimators," *SIAM J. Matrix Anal. Appl.* *13*, 90–101.

B. Kågström (1994). "A Perturbation Analysis of the Generalized Sylvester Equation $(AR - LB, DR - LE) = (C, F)$," *SIAM J. Matrix Anal. Appl.* *15*, 1045–1060.

E. Elmroth and B. Kågström (1996). "The Set of 2-by-3 Matrix Pencils–Kronecker Structure and their Transitions under Perturbations," *SIAM J. Matrix Anal.*, to appear.

A. Edelman, E. Elmroth, and B. Kågström (1996). "A Geometric Approach to Perturbation Theory of Matrices and Matrix Pencils," *SIAM J. Matrix Anal.*, to appear.

Chapter 8

The Symmetric Eigenvalue Problem

The symmetric eigenvalue problem with its rich mathematical structure is one of the most aesthetically pleasing problems in numerical linear algebra. We begin our presentation with a brief discussion of the mathematical properties that underlie this computation. In §8.2 and §8.3 we develop various power iterations eventually focusing on the symmetric QR algorithm.

In §8.4 we discuss Jacobi's method, one of the earliest matrix algorithms to appear in the literature. This technique is of current interest because it is amenable to parallel computation and because under certain circumstances it has superior accuracy.

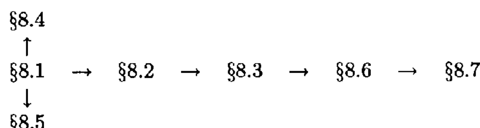
Various methods for the tridiagonal case are presented in §8.5. These include the method of bisection and a divide and conquer technique.

The computation of the singular value decomposition is detailed in §8.6. The central algorithm is a variant of the symmetric QR iteration that works on bidiagonal matrices.

In the final section we discuss the generalized eigenvalue problem $Ax = \lambda Bx$ for the important case when A is symmetric and B is symmetric positive definite. No suitable analog of the orthogonally-based QZ algorithm (see §7.7) exists for this specially structured, generalized eigenproblem. However, there are several successful methods that can be applied and these are presented along with a discussion of the generalized singular value decomposition.

Before You Begin

Chapter 1, §§2.1-2.5, and §2.7, Chapter 3, §§4.1-4.3, §§5.1-5.5 and §7.1.1 are assumed. Within this chapter there are the following dependencies:



Many of the algorithms and theorems in this chapter have unsymmetric counterparts in Chapter 7. However, except for a few concepts and definitions, our treatment of the symmetric eigenproblem can be studied before reading Chapter 7.

Complementary references include Wilkinson (1965), Stewart (1973), Gourlay and Watson (1973), Hager (1988), Chatelin (1993), Parlett (1980), Stewart and Sun (1990), Watkins (1991), Jennings and McKeowen (1992), and Datta (1995). Some Matlab functions important to this chapter are `schur` and `svd`. LAPACK connections include

LAPACK: Symmetric Eigenproblem	
_SYEV	All eigenvalues and vectors
_SYEVD	Same but uses divide and conquer for eigenvectors
_SYEVX	Selected eigenvalues and vectors
_SYTRD	Householder tridiagonalization
_SBTRD	Householder tridiagonalization (A banded)
_SPTRD	Householder tridiagonalization (A in packed storage)
_STEQR	All eigenvalues and vectors of tridiagonal by implicit QR
_STEDC	All eigenvalues and vectors of tridiagonal by divide and conquer
_STERF	All eigenvalues of tridiagonal by root-free QR
_PTEQR	All eigenvalues and eigenvectors of positive definite tridiagonal
_STEBZ	Selected eigenvalues of tridiagonal by bisection
_STEIN	Selected eigenvectors of tridiagonal by inverse iteration

LAPACK: Symmetric-Definite Eigenproblems	
_SYGST	Converts $A - \lambda B$ to $C - \lambda I$ form
_PBSTF	Split Cholesky factorization
_SBGST	Converts banded $A - \lambda B$ to $C - \lambda I$ form via split Cholesky

LAPACK: SVD	
_GESVD	$A = U\Sigma V^T$
_BDSQR	SVD of real bidiagonal matrix
_GEBRD	bidiagonalization of general matrix
_ORGBR	generates the orthogonal transformations
_GBBRD	bidiagonalization of band matrix

LAPACK: The Generalized Singular Value Problem	
_GGSVP	Converts $A^TA - \mu^2B^TB$ to triangular $A_1^TA_1 - \mu^2B_1^TB_1$
_TGSJA	Computes GSVD of a pair of triangular matrices.

8.1 Properties and Decompositions

In this section we set down the mathematics that is required to develop and analyze algorithms for the symmetric eigenvalue problem.

8.1.1 Eigenvalues and Eigenvectors

Symmetry guarantees that all of A's eigenvalues are real and that there is an orthonormal basis of eigenvectors.

Theorem 8.1.1 (Symmetric Schur Decomposition) *If $A \in \mathbb{R}^{n\times n}$ is symmetric, then there exists a real orthogonal Q such that*

$$Q^TAQ = \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n).$$

Moreover, for $k = 1{:}n$, $AQ(:,k) = \lambda_k Q(:,k)$. See Theorem 7.1.3.

Proof. Suppose $\lambda_1 \in \lambda(A)$ and that $x \in \mathbb{C}^n$ is a unit 2-norm eigenvector with $Ax = \lambda_1 x$. Since $\lambda_1 = x^HAx = x^HA^Hx = \overline{x^HAx} = \overline{\lambda_1}$ it follows that $\lambda_1 \in \mathbb{R}$. Thus, we may assume that $x \in \mathbb{R}^n$. Let $P_1 \in \mathbb{R}^{n\times n}$ be a Householder matrix such that $P_1^Tx = e_1 = I_n(:,1)$. It follows from $Ax = \lambda_1 x$ that $(P_1^TAP_1)e_1 = \lambda e_1$. This says that the first column of $P_1^TAP_1$ is a multiple of e_1. But since $P_1^TAP_1$ is symmetric it must have the form

$$P_1^TAP_1 = \begin{bmatrix} \lambda_1 & 0 \\ 0 & A_1 \end{bmatrix}$$

where $A_1 \in \mathbb{R}^{(n-1)\times(n-1)}$ is symmetric. By induction we may assume that there is an orthogonal $Q_1 \in \mathbb{R}^{(n-1)\times(n-1)}$ such that $Q_1^TA_1Q_1 = \Lambda_1$ is diagonal. The theorem follows by setting

$$Q = P_1\begin{bmatrix} 1 & 0 \\ 0 & Q_1 \end{bmatrix} \quad \text{and} \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \Lambda_1 \end{bmatrix}$$

and comparing columns in the matrix equation $AQ = Q\Lambda$. □

Example 8.1.1 If

$$A = \begin{bmatrix} 6.8 & 2.4 \\ 2.4 & 8.2 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} .6 & -.8 \\ .8 & .6 \end{bmatrix},$$

then Q is orthogonal and $Q^T AQ = \text{diag}(10,5)$.

For a symmetric matrix A we shall use the notation $\lambda_k(A)$ to designate the kth largest eigenvalue. Thus,

$$\lambda_n(A) \leq \cdots \leq \lambda_2(A) \leq \lambda_1(A).$$

It follows from the orthogonal invariance of the 2-norm that A has singular values $\{|\lambda_1(A)|, \ldots, |\lambda_n(A)|\}$ and so

$$\| A \|_2 = \max\{ |\lambda_1(A)| , |\lambda_n(A)| \}.$$

The eigenvalues of a symmetric matrix have a "minimax" characterization based on the values that can be assumed by the quadratic form ratio $x^T Ax/x^T x$.

Theorem 8.1.2 (Courant-Fischer Minimax Theorem) *If* $A \in \mathbb{R}^{n\times n}$ *is symmetric, then*

$$\lambda_k(A) = \max_{\dim(S)=k} \min_{0\neq y\in S} \frac{y^T Ay}{y^T y}$$

for $k = 1{:}n$.

Proof. Let $Q^T AQ = \text{diag}(\lambda_i)$ be the Schur decomposition with $\lambda_k = \lambda_k(A)$ and $Q = [\, q_1, q_2, \ldots, q_n \,]$. Define

$$S_k = \text{span}\{q_1, \ldots, q_k\},$$

the invariant subspace associated with $\lambda_1, \ldots, \lambda_k$. It is easy to show that

$$\max_{\dim(S)=k} \min_{0\neq y\in S} \frac{y^T Ay}{y^T y} \geq \min_{0\neq y\in S_k} \frac{y^T Ay}{y^T y} = q_k^T Aq_k = \lambda_k(A).$$

To establish the reverse inequality, let S be any k-dimensional subspace and note that it must intersect $\text{span}\{q_k, \ldots, q_n\}$, a subspace that has dimension $n - k + 1$. If $y_* = \alpha_k q_k + \cdots + \alpha_n q_n$ is in this intersection, then

$$\min_{0\neq y\in S} \frac{y^T Ay}{y^T y} \leq \frac{y_*^T Ay_*}{y_*^T y_*} \leq \lambda_k(A).$$

Since this inequality holds for all k-dimensional subspaces,

$$\max_{\dim(S)=k} \min_{0\neq y\in S} \frac{y^T Ay}{y^T y} \leq \lambda_k(A)$$

thereby completing the proof of the theorem. □

If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then $\lambda_n(A) > 0$.

8.1.2 Eigenvalue Sensitivity

An important solution framework for the symmetric eigenproblem involves the production of a sequence of orthogonal transformations $\{Q_k\}$ with the property that the matrices $Q_k^T A Q_k$ are progressively "more diagonal." The question naturally arises, how well do the diagonal elements of a matrix approximate its eigenvalues?

Theorem 8.1.3 (Gershgorin) *Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and that $Q \in \mathbb{R}^{n\times n}$ is orthogonal. If $Q^T A Q = D + F$ where $D = \text{diag}(d_1, \ldots, d_n)$ and F has zero diagonal entries, then*

$$\lambda(A) \subseteq \bigcup_{i=1}^{n} [d_i - r_i, d_i + r_i]$$

where $r_i = \sum_{j=1}^{n} |f_{ij}|$ for $i = 1{:}n$. See Theorem 7.2.1.

Proof. Suppose $\lambda \in \lambda(A)$ and assume without loss of generality that $\lambda \neq d_i$ for $i = 1{:}n$. Since $(D - \lambda I) + F$ is singular, it follows from Lemma 2.3.3 that

$$1 \le \| (D - \lambda I)^{-1} F \|_\infty = \sum_{j=1}^{n} \frac{|f_{kj}|}{|d_k - \lambda|} = \frac{r_k}{|d_k - \lambda|}$$

for some k, $1 \le k \le n$. But this implies that $\lambda \in [d_k - r_k, d_k + r_k]$. □

Example 8.1.2 The matrix

$$A = \begin{bmatrix} 2.0000 & 0.1000 & 0.2000 \\ 0.2000 & 5.0000 & 0.3000 \\ 0.1000 & 0.3000 & -1.0000 \end{bmatrix}$$

has Gerschgorin intervals $[1.7, 2.3]$, $[4.5, 5.5]$, and $[-1.4, -.6]$ and eigenvalues 1.9984, 5.0224, and -1.0208.

The next results show that if A is perturbed by a symmetric matrix E, then its eigenvalues do not move by more than $\| E \|$.

Theorem 8.1.4 (Wielandt-Hoffman) *If A and $A + E$ are n-by-n symmetric matrices, then*

$$\sum_{i=1}^{n} (\lambda_i(A + E) - \lambda_i(A))^2 \le \| E \|_F^2 .$$

Proof. A proof can be found in Wilkinson (1965, pp.104–8) or Stewart and Sun (1991, pp.189–191). See also P8.1.5. □

Example 8.1.3 If

$$A = \begin{bmatrix} 6.8 & 2.4 \\ 2.4 & 8.2 \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} .002 & .003 \\ .003 & .001 \end{bmatrix},$$

then $\lambda(A) = \{5, 10\}$ and $\lambda(A+E) = \{4.9988, 10.004\}$ confirming that

$$1.95 \times 10^{-5} = |4.9988 - 5|^2 + |10.004 - 10|^2 \leq \| E \|_F^2 = 2.3 \times 10^{-5}.$$

Theorem 8.1.5 *If A and $A + E$ are n-by-n symmetric matrices, then*

$$\lambda_k(A) + \lambda_n(E) \leq \lambda_k(A+E) \leq \lambda_k(A) + \lambda_1(E) \qquad k = 1{:}n.$$

Proof. This follows from the minimax characterization. See Wilkinson (1965, pp.101–2) or Stewart and Sun (1990, p.203). □

Example 8.1.4 If

$$A = \begin{bmatrix} 6.8 & 2.4 \\ 2.4 & 8.2 \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} .002 & .003 \\ .003 & .001 \end{bmatrix},$$

then $\lambda(A) = \{5, 10\}$, $\lambda(E) = \{-.0015, .0045\}$, and $\lambda(A+E) = \{4.9988, 10.0042\}$. confirming that

$$\begin{array}{lcl} 5 - .0015 & \leq 4.9988 \leq & 5 + .0045 \\ 10 - .0015 & \leq 10.0042 \leq & 10 + .0045. \end{array}$$

Corollary 8.1.6 *If A and $A + E$ are n-by-n symmetric matrices, then*

$$|\lambda_k(A+E) - \lambda_k(A)| \leq \| E \|_2$$

for $k = 1{:}n$.

Proof.

$$|\lambda_k(A+E) - \lambda_k(A)| \leq \max\{|\lambda_n(E)|, |\lambda_1(E)|\} = \| E \|_2. \ \square$$

Several more useful perturbation results follow from the minimax property.

Theorem 8.1.7 (Interlacing Property) *If $A \in \mathbb{R}^{n \times n}$ is symmetric and $A_r = A(1{:}r, 1{:}r)$, then*

$$\lambda_{r+1}(A_{r+1}) \leq \lambda_r(A_r) \leq \lambda_r(A_{r+1}) \leq \cdots \leq \lambda_2(A_{r+1}) \leq \lambda_1(A_r) \leq \lambda_1(A_{r+1})$$

for $r = 1{:}n-1$.

Proof. Wilkinson (1965, pp.103–4). □

Example 8.1.5 If

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$$

then $\lambda(A_1) = \{1\}$, $\lambda(A_2) = \{.3820,\ 2.6180\}$, $\lambda(A_3) = \{.1270,\ 1.0000,\ 7.873\}$, and $\lambda(A_4) = \{.0380,\ .4538,\ 2.2034, 26.3047\}$.

Theorem 8.1.8 *Suppose* $B = A + \tau cc^T$ *where* $A \in \mathbb{R}^{n\times n}$ *is symmetric,* $c \in \mathbb{R}^n$ *has unit 2-norm and* $\tau \in \mathbb{R}$. *If* $\tau \geq 0$, *then*

$$\lambda_i(B) \in [\lambda_i(A),\ \lambda_{i-1}(A)] \qquad i = 2{:}n$$

while if $\tau \leq 0$ *then*

$$\lambda_i(B) \in [\lambda_{i+1}(A), \lambda_i(A)], \qquad i = 1{:}n-1\,.$$

In either case, there exist nonnegative $m_1, \ldots, m_n$ *such that*

$$\lambda_i(B) = \lambda_i(A) + m_i\tau, \qquad i = 1{:}n$$

with $m_1 + \cdots + m_n = 1$.

Proof. Wilkinson (1965, pp.94–97). See also P8.1.8. □

8.1.3 Invariant Subspaces

Many eigenvalue computations proceed by breaking the original problem into a collection of smaller subproblems. The following result is the basis for this solution framework.

Theorem 8.1.9 *Suppose* $A \in \mathbb{R}^{n\times n}$ *is symmetric and that*

$$Q = [\ \underset{r}{Q_1} \quad \underset{n-r}{Q_2}\ ]$$

is orthogonal. If $\operatorname{ran}(Q_1)$ *is an invariant subspace, then*

$$Q^TAQ = D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} \qquad (8.1.1)$$
$$\qquad\qquad\qquad\qquad \begin{matrix} r & n-r \end{matrix}$$

and $\lambda(A) = \lambda(D_1) \cup \lambda(D_2)$. *See also Lemma 7.1.2.*

Proof. If

$$Q^TAQ = \begin{bmatrix} D_1 & E_{21}^T \\ E_{21} & D_2 \end{bmatrix},$$

then from $AQ = QD$ we have $AQ_1 - Q_1D_1 = Q_2E_{21}$. Since $\text{ran}(Q_1)$ is invariant, the columns of Q_2E_{21} are also in $\text{ran}(Q_1)$ and therefore perpendicular to the columns of Q_2. Thus,

$$0 = Q_2^T(AQ_1 - Q_1D_1) = Q_2^TQ_2E_{21} = E_{21}.$$

and so (8.1.1) holds. It is easy to show

$$\det(A - \lambda I_n) = \det(Q^TAQ - \lambda I_n) = \det(D_1 - \lambda I_r)\det(D_2 - \lambda I_{n-r})$$

confirming that $\lambda(A) = \lambda(D_1) \cup \lambda(D_2)$. □

The sensitivity to perturbation of an invariant subspace depends upon the separation of the associated eigenvalues from the rest of the spectrum. The appropriate measure of separation between the eigenvalues of two symmetric matrices B and C is given by

$$\text{sep}(B, C) = \min_{\substack{\lambda \in \lambda(B) \\ \mu \in \lambda(C)}} |\lambda - \mu|. \tag{8.1.2}$$

With this definition we have

Theorem 8.1.10 *Suppose A and $A + E$ are n-by-n symmetric matrices and that*

$$Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}_{\;r \quad n-r}$$

is an orthogonal matrix such that $\text{ran}(Q_1)$ is an invariant subspace for A. Partition the matrices Q^TAQ and Q^TEQ as follows:

$$Q^TAQ = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} \qquad Q^TEQ = \begin{bmatrix} E_{11} & E_{21}^T \\ E_{21} & E_{22} \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix}.$$
$$\qquad\quad r \quad n-r \qquad\qquad\qquad\qquad r \quad n-r$$

If $\text{sep}(D_1, D_2) > 0$ and

$$\| E \|_2 \le \frac{\text{sep}(D_1, D_2)}{5},$$

then there exists a matrix $P \in \mathbb{R}^{(n-r)\times r}$ with

$$\| P \|_2 \le \frac{4}{\text{sep}(D_1, D_2)} \| E_{21} \|_2$$

such that the columns of $\hat{Q}_1 = (Q_1 + Q_2P)(I + P^TP)^{-1/2}$ define an orthonormal basis for a subspace that is invariant for $A+E$. See also Theorem 7.2.4.

Proof. This result is a slight adaptation of of Theorem 4.11 in Stewart (1973). The matrix $(I + P^TP)^{-1/2}$ is the inverse of the square root of $I + P^TP$. See §4.2.10. □

Corollary 8.1.11 *If the conditions of the theorem hold, then*

$$\text{dist}(\text{ran}(Q_1), \text{ran}(\hat{Q}_1)) \leq \frac{4}{\text{sep}(D_1, D_2)} \| E_{21} \|_2.$$

See also Corollary 7.2.5.

Proof. It can be shown using the SVD that

$$\| P(I + P^TP)^{-1/2} \|_2 \leq \| P \|_2. \tag{8.1.3}$$

Since $Q_2^T\hat{Q}_1 = P(I + P^HP)^{-1/2}$ it follows that

$$\begin{aligned} \text{dist}(\text{ran}(Q_1), \text{ran}(\hat{Q}_1)) &= \| Q_2^T\hat{Q}_1 \|_2 = \| P(I + P^HP)^{-1/2} \|_2 \\ &\leq \| P \|_2 \leq \| E_{21} \|_2/\text{sep}(D_1, D_2). \ \square \end{aligned}$$

Thus, the reciprocal of $\text{sep}(D_1, D_2)$ can be thought of as a condition number that measures the sensitivity of $\text{ran}(Q_1)$ as an invariant subspace.

The effect of perturbations on a single eigenvector is sufficiently important that we specialize the above results to this important case.

Theorem 8.1.12 *Suppose A and $A + E$ are n-by-n symmetric matrices and that*

$$Q = \begin{array}{cc} [\ q_1 & Q_2\ ] \\ 1 & n-1 \end{array}$$

is an orthogonal matrix such that q_1 is an eigenvector for A. Partition the matrices Q^TAQ and Q^TEQ as follows:

$$Q^TAQ = \begin{array}{c} \left[\begin{array}{cc} \lambda & 0 \\ 0 & D_2 \end{array}\right] \\ \begin{array}{cc} 1 & n-1 \end{array} \end{array} \begin{array}{c} 1 \\ n-1 \end{array} \qquad Q^TEQ = \begin{array}{c} \left[\begin{array}{cc} \epsilon & e^T \\ e & E_{22} \end{array}\right] \\ \begin{array}{cc} 1 & n-1 \end{array} \end{array} \begin{array}{c} 1 \\ n-1 \end{array} .$$

If $d = \min_{\mu \in \lambda(D_2)} |\lambda - \mu| > 0$ *and*

$$\| E \|_2 \leq \frac{d}{4},$$

then there exists $p \in \mathbb{R}^{n-1}$ satisfying

$$\| p \|_2 \leq \frac{4}{d} \| e \|_2$$

such that $\hat{q}_1 = (q_1 + Q_2 p)/\sqrt{1 + p^T p}$ *is a unit 2-norm eigenvector for* $A+E$. *Moreover,*

$$\text{dist}(\text{span}\{q_1\}, \text{span}\{\hat{q}_1\}) \;=\; \sqrt{1 - (q_1^T \hat{q}_1)^2} \;\le\; \frac{4}{d} \| e \|_2.$$

See also Corollary 7.2.6.

Proof. Apply Theorem 8.1.10 and Corollary 8.1.11 with $r = 1$ and observe that if $D_1 = (\lambda)$, then $d = \text{sep}(D_1, D_2)$. □

Example 8.1.6 If $A = \text{diag}(.999,\ 1.001,\ 2.)$, and

$$E \;=\; \begin{bmatrix} 0.00 & 0.01 & 0.01 \\ 0.01 & 0.00 & 0.01 \\ 0.01 & 0.01 & 0.00 \end{bmatrix},$$

then $\hat{Q}^T(A + E)\hat{Q} = \text{diag}(.9899,\ 1.0098,\ 2.0002)$ where

$$\hat{Q} \;=\; \begin{bmatrix} -.7418 & .6706 & .0101 \\ .6708 & .7417 & .0101 \\ .0007 & -.0143 & .9999 \end{bmatrix}$$

is orthogonal. Let $\hat{q}_i = \hat{Q}e_i$, $i = 1, 2, 3$. Thus, $\hat{q}_i$ is the perturbation of A's eigenvector $q_i = e_i$. A calculation shows that

$$\text{dist}\{\text{span}\{q_1\}, \text{span}\{\hat{q}_1\}\} \;=\; \text{dist}\{\text{span}\{q_2\}, \text{span}\{\hat{q}_2\}\} \;=\; .67$$

Thus, because they are associated with nearby eigenvalues, the eigenvectors q_1 and q_2 cannot be computed accurately. On the other hand, since λ_1 and λ_2 are well separated from λ_3, they define a two-dimensional subspace that is not particularly sensitive as $\text{dist}\{\text{span}\{q_1, q_2\}, \text{span}\{\hat{q}_1, \hat{q}_2\}\} = .01$.

8.1.4 Approximate Invariant Subspaces

If the columns of $Q_1 \in \mathbb{R}^{n \times r}$ are independent and the *residual matrix* $R = AQ_1 - Q_1 S$ is small for some $S \in \mathbb{R}^{r \times r}$, then the columns of Q_1 define an approximate invariant subspace. Let us discover what we can say about the eigensystem of A when in the possession of such a matrix.

Theorem 8.1.13 *Suppose* $A \in \mathbb{R}^{n \times n}$ *and* $S \in \mathbb{R}^{r \times r}$ *are symmetric and that*

$$AQ_1 - Q_1 S \;=\; E_1$$

where $Q_1 \in \mathbb{R}^{n \times r}$ *satisfies* $Q_1^T Q_1 = I_r$. *Then there exist* $\mu_1, \ldots, \mu_r \in \lambda(A)$ *such that*

$$|\mu_k - \lambda_k(S)| \;\le\; \sqrt{2} \, \| E_1 \|_2$$

for $k = 1{:}r$.

Proof. Let $Q_2 \in \mathbb{R}^{n\times(n-r)}$ be any matrix such that $Q = [\ Q_1,\ Q_2\ ]$ is orthogonal. It follows that

$$Q^TAQ = \begin{bmatrix} S & 0 \\ 0 & Q_2^TAQ_2 \end{bmatrix} + \begin{bmatrix} Q_1^TE_1 & E_1^TQ_2 \\ Q_2^TE_1 & 0 \end{bmatrix} \equiv B + E$$

and so by using Corollary 8.1.6 we have $|\lambda_k(A) - \lambda_k(B)| \le \|E\|_2$ for $k = 1{:}n$. Since $\lambda(S) \subseteq \lambda(B)$, there exist $\mu_1, \ldots, \mu_r \in \lambda(A)$ such that

$$|\mu_k - \lambda_k(S)| \le \|E\|_2$$

for $k = 1{:}r$. The theorem follows by noting that for any $x \in \mathbb{R}^r$ and $y \in \mathbb{R}^{n-r}$ we have

$$\left\| E\begin{bmatrix} x \\ y \end{bmatrix} \right\|_2 \le \|E_1x\|_2 + \|E_1^TQ_2y\|_2 \le \|E_1\|_2\|x\|_2 + \|E_1\|_2\|y\|_2$$

from which we readily conclude that $\|E\|_2 \le \sqrt{2}\|E_1\|_2$. □

Example 8.1.7 If

$$A = \begin{bmatrix} 6.8 & 2.4 \\ 2.4 & 8.2 \end{bmatrix}, \qquad Q_1 = \begin{bmatrix} .7994 \\ .6007 \end{bmatrix}, \text{ and } S = (5.1) \in \mathbb{R}$$

then

$$AQ_1 - Q_1S = \begin{bmatrix} -.0828 \\ -.0562 \end{bmatrix} = E_1.$$

The theorem predicts that A has an eigenvalue within $\sqrt{2}\,\|E_1\|_2 \approx .1415$ of 5.1. This is true since $\lambda(A) = \{5, 10\}$.

The eigenvalue bounds in Theorem 8.1.13 depend on $\|AQ_1 - Q_1S\|_2$. Given A and Q_1, the following theorem indicates how to choose S so that this quantity is minimized in the Frobenius norm.

Theorem 8.1.14 *If $A \in \mathbb{R}^{n\times n}$ is symmetric and $Q_1 \in \mathbb{R}^{n\times r}$ has orthonormal columns, then*

$$\min_{S\in\mathbb{R}^{r\times r}} \|AQ_1 - Q_1S\|_F = \|(I - Q_1Q_1^T)AQ_1\|_F$$

and $S = Q_1^TAQ_1$ is the minimizer.

Proof. Let $Q_2 \in \mathbb{R}^{n\times(n-r)}$ be such that $Q = [\ Q_1,\ Q_2\ ]$ is orthogonal. For any $S \in \mathbb{R}^{r\times r}$ we have

$$\begin{aligned} \|AQ_1 - Q_1S\|_F^2 &= \|Q^TAQ_1 - Q^TQ_1S\|_F^2 \\ &= \|Q_1^TAQ_1 - S\|_F^2 + \|Q_2^TAQ_1\|_F^2. \end{aligned}$$

Clearly, the minimizing S is given by $S = Q_1^T A Q_1$. □

This result enables us to associate any r-dimensional subspace ran(Q_1), with a set of r "optimal" eigenvalue-eigenvector approximates.

Theorem 8.1.15 *Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and that $Q_1 \in \mathbb{R}^{n\times r}$ satisfies $Q_1^T Q_1 = I_r$. If*

$$Z^T(Q_1^T A Q_1)Z \;=\; \text{diag}(\theta_1,\ldots,\theta_r) \;=\; D$$

is the Schur decomposition of $Q_1^T A Q_1$ and $Q_1 Z = [\, y_1,\ldots,y_r \,]$, then

$$\| \, Ay_k - \theta_k y_k \, \|_2 \;=\; \| \, (I - Q_1 Q_1^T)AQ_1 Z e_k \, \|_2 \;\le\; \| \, (I - Q_1 Q_1^T)AQ_1 \, \|_2$$

for $k = 1{:}r$.

Proof.

$$Ay_k - \theta_k y_k \;=\; AQ_1 Z e_k = Q_1 Z D e_k \;=\; (AQ_1 - Q_1(Q_1^T A Q_1))Ze_k.$$

The theorem follows by taking norms. □

In Theorem 8.1.15, the θ_k are called *Ritz values*, the y_k are called *Ritz vectors*, and the (θ_k, y_k) are called *Ritz pairs*.

The usefulness of Theorem 8.1.13 is enhanced if we weaken the assumption that the columns of Q_1 are orthonormal. As can be expected, the bounds deteriorate with the loss of orthogonality.

Theorem 8.1.16 *Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and that*

$$AX_1 - X_1 S \;=\; F_1,$$

where $X_1 \in \mathbb{R}^{n\times r}$ and $S = X_1^T A X_1$. If

$$\| \, X_1^T X_1 - I_r \, \|_2 = \tau < 1, \tag{8.1.4}$$

then there exist $\mu_1,\ldots,\mu_r \in \lambda(A)$ such that

$$|\mu_k - \lambda_k(S)| \;\le\; \sqrt{2}\,(\| \, F_1 \, \|_2 \;+\; \tau(2+\tau)\| \, A \, \|_2)$$

for $k = 1{:}r$.

Proof. Let $X_1 = ZP$ be the polar decomposition of X_1. Recall from §4.2.10 that this means $Z \in \mathbb{R}^{n\times r}$ has orthonormal columns and $P \in \mathbb{R}^{k\times k}$ is a symmetric positive semidefinite matrix that satisfies $P^2 = X_1^T X_1$. Taking norms in the equation

$$\begin{aligned} E_1 \equiv AZ - ZS \;&=\; (AX_1 - X_1 S) \;+\; A(Z - X_1) \;-\; (Z - X_1)S \\ &=\; F_1 \;+\; AZ(I - P) \;-\; Z(I - P)X_1^T A X_1 \end{aligned}$$

gives

$$\| E_1 \|_2 \le \| F_1 \|_2 + \| A \|_2 \| I - P \|_2 \left(1 + \| X_1 \|_2^2 \right). \tag{8.1.5}$$

Equation (8.1.4) implies that

$$\| X_1 \|_2^2 \le 1 + \tau. \tag{8.1.6}$$

Since P is positive semidefinite, $(I + P)$ is nonsingular and so

$$I - P = (I + P)^{-1}(I - P^2) = (I + P)^{-1}(I - X_1^T X_1)$$

which implies $\| I - P \|_2 \le \tau$. By substituting this inequality and (8.1.6) into (8.1.5) we have $\| E_1 \|_2 \le \| F_1 \|_2 + \tau(2 + \tau)\| A \|_2$. The proof is completed by noting that we can use Theorem 8.1.13 with $Q_1 = Z$ to relate the eigenvalues of A and S via the residual E_1. □

8.1.5 The Law of Inertia

The *inertia* of a symmetric matrix A is a triplet of nonnegative integers (m, z, p) where m, z, and p are respectively the number of negative, zero, and positive elements of $\lambda(A)$.

Theorem 8.1.17 (Sylvester Law of Inertia) *If $A \in \mathbb{R}^{n\times n}$ is symmetric and $X \in \mathbb{R}^{n\times n}$ is nonsingular, then A and X^TAX have the same inertia.*

Proof. Suppose for some r that $\lambda_r(A) > 0$ and define the subspace $S_0 \subseteq \mathbb{R}^n$ by

$$S_0 = \text{span}\{X^{-1}q_1, \ldots, X^{-1}q_r\}, \qquad q_i \ne 0$$

where $Aq_i = \lambda_i(A)q_i$ and $i = 1{:}r$. From the minimax characterization of $\lambda_r(X^TAX)$ we have

$$\lambda_r(X^TAX) = \max_{\dim(S)=r} \; \min_{y\in S} \frac{y^T(X^TAX)y}{y^Ty} \ge \min_{y\in S_0} \frac{y^T(X^TAX)y}{y^Ty}.$$

Since

$$y \in \mathbb{R}^n \;\Rightarrow\; \frac{y^T(X^TX)y}{y^Ty} \ge \sigma_n(X)^2$$

$$y \in S_0 \;\Rightarrow\; \frac{y^T(X^TAX)y}{y^Ty} \ge \lambda_r(A)$$

it follows that

$$\lambda_r(X^TAX) \ge \min_{y\in S_0} \left\{ \frac{y^T(X^TAX)y}{y^T(X^TX)y} \frac{y^T(X^TX)y}{y^Ty} \right\} \ge \lambda_r(A)\sigma_n(X)^2.$$

An analogous argument with the roles of A and X^TAX reversed shows that

$$\lambda_r(A) \geq \lambda_r(X^TAX)\sigma_n(X^{-1})^2 = \frac{\lambda_r(X^TAX)}{\sigma_1(X)^2}.$$

Thus, $\lambda_r(A)$ and $\lambda_r(X^TAX)$ have the same sign and so we have shown that A and X^TAX have the same number of positive eigenvalues. If we apply this result to $-A$, we conclude that A and X^TAX have the same number of negative eigenvalues. Obviously, the number of zero eigenvalues possessed by each matrix is also the same. □

Example 8.1.8 If $A = \text{diag}(3, 2, -1)$ and

$$X = \begin{bmatrix} 1 & 4 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix},$$

then

$$X^TAX = \begin{bmatrix} 3 & 12 & 15 \\ 12 & 50 & 64 \\ 15 & 64 & 82 \end{bmatrix}$$

and $\lambda(X^TAX) = \{134.769, \ .3555, \ -.1252\}$.

Problems

P8.1.1 Without using any of the results in this section, show that the eigenvalues of a 2-by-2 symmetric matrix must be real.

P8.1.2 Compute the Schur decomposition of $A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$.

P8.1.3 Show that the eigenvalues of a Hermitian matrix ($A^H = A$) are real. For each theorem and corollary in this section, state and prove the corresponding result for Hermitian matrices. Which results have analogs when A is skew-symmetric? (Hint: If $A^T = -A$, then iA is Hermitian.)

P8.1.4 Show that if $X \in \mathbb{R}^{n \times r}$, $r \leq n$, and $\| X^TX - I \| = \tau < 1$, then $\sigma_{min}(X) \geq 1-\tau$.

P8.1.5 Suppose $A, E \in \mathbb{R}^{n \times n}$ are symmetric and consider the Schur decomposition $A + tE = QDQ^T$ where we *assume* that $Q = Q(t)$ and $D = D(t)$ are continuously differentiable functions of $t \in \mathbb{R}$. Show that $\dot{D}(t) = \text{diag}(Q(t)^TEQ(t))$ where the matrix on the right is the diagonal part of $Q(t)^TEQ(t)$. Establish the Wielandt-Hoffman theorem by integrating both sides of this equation from 0 to 1 and taking Frobenius norms to show that

$$\| D(1) - D(0) \|_F \leq \int_0^1 \| \text{diag}(Q(t)^TEQ(t) \|_F dt \leq \| E \|_F.$$

P8.1.6 Prove Theorem 8.1.5.

P8.1.7 Prove Theorem 8.1.7.

P8.1.8 If $C \in \mathbb{R}^{n \times n}$ then the *trace function* $\text{tr}(C) = c_{11} + \cdots + c_{nn}$ equals the sum of C's eigenvalues. Use this to prove Theorem 8.1.8.

P8.1.9 Show that if $B \in \mathbb{R}^{m \times m}$ and $C \in \mathbb{R}^{n \times n}$ are symmetric, then $\text{sep}(B, C) = \min$

$\| BX - XC \|_F$ where the min is taken over all matrices in $\mathbb{R}^{m\times n}$.

P8.1.10 Prove the inequality (8.1.3).

P8.1.11 Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and $C \in \mathbb{R}^{n\times r}$ has full column rank and assume that $r \ll n$. By using Theorem 8.1.8 relate the eigenvalues of $A + CC^T$ to the eigenvalues of A.

Notes and References for Sec. 8.1

The perturbation theory for the symmetric eigenvalue problem is surveyed in Wilkinson (1965, chapter 2), Parlett (1980, chapters 10 and 11), and Stewart and Sun (1990, chapters 4 and 5). Some representative papers in this well-researched area include

G.W. Stewart (1973). "Error and Perturbation Bounds for Subspaces Associated with Certain Eigenvalue Problems," *SIAM Review 15*, 727–64.

C.C. Paige (1974). "Eigenvalues of Perturbed Hermitian Matrices," *Lin. Alg. and Its Applic* . 8, 1–10.

A. Ruhe (1975). "On the Closeness of Eigenvalues and Singular Values for Almost Normal Matrices," *Lin. Alg. and Its Applic. 11*, 87–94.

W. Kahan (1975). "Spectra of Nearly Hermitian Matrices," *Proc. Amer. Math. Soc. 48*, 11 17.

A. Schonhage (1979). "Arbitrary Perturbations of Hermitian Matrices," *Lin. Alg. and Its Applic. 24*, 143–49.

P. Deift, T. Nanda, and C. Tomei (1983). "Ordinary Differential Equations and the Symmetric Eigenvalue Problem," *SIAM J. Numer. Anal. 20*, 1–22.

D.S. Scott (1985). "On the Accuracy of the Gershgorin Circle Theorem for Bounding the Spread of a Real Symmetric Matrix," *Lin. Alg. and Its Applic. 65*, 147–155

J.-G. Sun (1995). "A Note on Backward Error Perturbations for the Hermitian Eigenvalue Problem," *BIT 35*, 385–393.

R.-C. Li (1996). "Relative Perturbation Theory (I) Eigenvalue and Singular Value Variations," Technical Report UCB//CSD-94-855, Department of EECS, University of California at Berkeley.

R.-C. Li (1996). "Relative Perturbation Theory (II) Eigenspace and Singular Subspace Variations," Technical Report UCB//CSD-94-856, Department of EECS, University of California at Berkeley.

8.2 Power Iterations

Assume that $A \in \mathbb{R}^{n\times n}$ is symmetric and that $U_0 \in \mathbb{R}^{n\times n}$ is orthogonal. Consider the following *QR iteration*:

$$
\begin{array}{l}
T_0 = U_0^T A U_0 \\
\textbf{for } k = 1, 2, \ldots \\
\qquad T_{k-1} = U_k R_k \quad \text{(QR factorization)} \\
\qquad T_k = R_k U_k \\
\textbf{end}
\end{array}
\tag{8.2.1}
$$

Since $T_k = R_k U_k = U_k^T (U_k R_k) U_k = U_k^T T_{k-1} U_k$ it follows by induction that

$$T_k = (U_0 U_1 \cdots U_k)^T A (U_0 U_1 \cdots U_k). \tag{8.2.2}$$

Thus, each T_k is orthogonally similar to A. Moreover, the T_k almost always converge to diagonal form and so it can be said that (8.2.1) almost always "converges" to a Schur decomposition of A. In order to establish this remarkable result we first consider the power method and the method of orthogonal iteration.

8.2.1 The Power Method

Given a unit 2-norm $q^{(0)} \in \mathbb{R}^n$, the *power method* produces a sequence of vectors $q^{(k)}$ as follows:

$$\begin{aligned}
&\textbf{for } k = 1, 2, \ldots \\
&\quad z^{(k)} = Aq^{(k-1)} \\
&\quad q^{(k)} = z^{(k)} / \| z^{(k)} \|_2 \\
&\quad \lambda^{(k)} = [q^{(k)}]^T A q^{(k)} \\
&\textbf{end}
\end{aligned} \tag{8.2.3}$$

If $q^{(0)}$ is not "deficient" and A's eigenvalue of maximum modulus is unique, then the $q^{(k)}$ converge to an eigenvector.

Theorem 8.2.1 *Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and that*

$$Q^T AQ = \text{diag}(\lambda_1, \ldots, \lambda_n)$$

where $Q = [\, q_1, \ldots, q_n \,]$ is orthogonal and $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Let the vectors q_k be specified by (8.2.3) and define $\theta_k \in [0, \pi/2]$ by

$$\cos(\theta_k) = \left| q_1^T q^{(k)} \right| .$$

If $\cos(\theta_0) \neq 0$, then

$$|\sin(\theta_k)| \;\leq\; \tan(\theta_0) \left| \frac{\lambda_2}{\lambda_1} \right|^k \tag{8.2.4}$$

$$|\lambda^{(k)} - \lambda| \;\leq\; |\lambda_1 - \lambda_n| \tan(\theta_0)^2 \left| \frac{\lambda_2}{\lambda_1} \right|^{2k} . \tag{8.2.5}$$

Proof. From the definition of the iteration, it follows that $q^{(k)}$ is a multiple of $A^k q^{(0)}$ and so

$$|\sin(\theta_k)|^2 \;=\; 1 \;-\; \left(q_1^T q^{(k)} \right)^2 \;=\; 1 \;-\; \left(\frac{q_1^T A^k q^{(0)}}{\| A^k q^{(0)} \|_2} \right)^2 .$$

If $q^{(0)}$ has the eigenvector expansion $q^{(0)} = a_1 q_1 + \cdots + a_n q_n$, then

$$|a_1| = |q_1^T q^{(0)}| = \cos(\theta_0) \neq 0,$$

$$a_1^2 + \cdots + a_n^2 = 1,$$

and

$$A^k q^{(0)} = a_1\lambda_1^k q_1 + a_2\lambda_2^k q_2 + \cdots + a_n\lambda_n^k q_n \,.$$

Thus,

$$|\sin(\theta_k)|^2 = 1 - \frac{a_1^2\lambda_1^{2k}}{\sum_{i=1}^{n} a_i^2\lambda_i^{2k}} = \frac{\sum_{i=2}^{n} a_i^2\lambda_i^{2k}}{\sum_{i=1}^{n} a_i^2\lambda_i^{2k}}$$

$$\le \frac{\sum_{i=2}^{n} a_i^2\lambda_i^{2k}}{a_1^2\lambda_1^{2k}} = \frac{1}{a_1^2}\sum_{i=2}^{n} a_i^2\left(\frac{\lambda_i}{\lambda_1}\right)^{2k}$$

$$\le \frac{1}{a_1^2}\left(\sum_{i=2}^{n} a_i^2\right)\left(\frac{\lambda_2}{\lambda_1}\right)^{2k} = \frac{1-a_1^2}{a_1^2}\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}$$

$$= \tan(\theta_0)^2\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}.$$

This proves (8.2.4). Likewise,

$$\lambda^{(k)} = \left[q^{(k)}\right]^T A q^{(k)} = \frac{\left[q^{(0)}\right]^T A^{2k+1} q^{(0)}}{\left[q^{(0)}\right]^T A^{2k} q^{(0)}} = \frac{\sum_{i=1}^{n} a_i^2\lambda_i^{2k+1}}{\sum_{i=1}^{n} a_i^2\lambda_i^{2k}}$$

and so

$$\left|\lambda^{(k)} - \lambda_1\right| = \left|\frac{\sum_{i=2}^{n} a_i^2\lambda_i^{2k}(\lambda_i - \lambda_1)}{\sum_{i=1}^{n} a_i^2\lambda_i^{2k}}\right| \le |\lambda_1 - \lambda_n|\frac{1}{a_1^2}\sum_{i=2}^{n} a_i^2\left(\frac{\lambda_i}{\lambda_1}\right)^{2k}$$

$$\le |\lambda_1 - \lambda_n|\tan(\theta_0)^2\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}. \square$$

Example 8.2.1 The eigenvalues of

$$A = \begin{bmatrix} -1.6407 & 1.0814 & 1.2014 & 1.1539 \\ 1.0814 & 4.1573 & 7.4035 & -1.0463 \\ 1.2014 & 7.4035 & 2.7890 & -1.5737 \\ 1.1539 & -1.0463 & -1.5737 & 8.6944 \end{bmatrix}$$

are given by $\lambda(A) = \{12, 8, -4, -2\}$. If (8.2.3) is applied to this matrix with $q^{(0)} = [1\ 0\ 0\ 0]^T$, then

k	$\lambda^{(k)}$
1	2.3156
2	8.6802
3	10.3163
4	11.0663
5	11.5259
6	11.7747
7	11.8967
8	11.9534
9	11.9792
10	11.9907

Observe the convergence to $\lambda_1 = 12$ with rate $|\lambda_2/\lambda_1|^{2k} = (8/12)^{2k} = (4/9)^k$.

Computable error bounds for the power method can be obtained by using Theorem 8.1.13. If

$$\| Aq^{(k)} - \lambda^{(k)} q^{(k)} \|_2 = \delta,$$

then there exists $\lambda \in \lambda(A)$ such that $|\lambda^{(k)} - \lambda| \leq \sqrt{2}\delta$.

8.2.2 Inverse Iteration

Suppose the power method is applied with A replaced by $(A - \lambda I)^{-1}$. If λ is very close to a distinct eigenvalue of A, then the next iterate vector will be very rich in the corresponding eigendirection:

$$\left.\begin{array}{l} x = \displaystyle\sum_{i=1}^{n} a_i q_i \\[2ex] Aq_i = \lambda_i q_i,\ i = 1{:}n \end{array}\right\} \Rightarrow (A - \lambda I)^{-1} x = \sum_{i=1}^{n} \frac{a_i}{\lambda_i - \lambda} q_i.$$

Thus, if $\lambda \approx \lambda_j$ and a_j is not too small, then this vector has a strong component in the direction of q_j. This process is called *inverse iteration* and it requires the solution of a linear system with matrix of coefficients $A - \lambda I$.

8.2.3 Rayleigh Quotient Iteration

Suppose $A \in \mathbb{R}^{n \times n}$ is symmetric and that x is a given nonzero n-vector. A simple differentiation reveals that

$$\lambda = r(x) \equiv \frac{x^T A x}{x^T x}$$

minimizes $\| (A - \lambda I)x \|_2$. (See also Theorem 8.1.14.) The scalar $r(x)$ is called the *Rayleigh quotient* of x. Clearly, if x is an approximate eigenvector, then r(x) is a reasonable choice for the corresponding eigenvalue.

Combining this idea with inverse iteration gives rise to the *Rayleigh quotient iteration:*

$$
\begin{array}{l}
x_0 \text{ given, } \| x_0 \|_2 = 1 \\
\textbf{for } k = 0, 1, \ldots \\
\quad \mu_k = r(x_k) \\
\quad \text{Solve } (A - \mu_k I) z_{k+1} = x_k \text{ for } z_{k+1} \\
\quad x_{k+1} = z_{k+1} / \| z_{k+1} \|_2 \\
\textbf{end}
\end{array}
\qquad (8.2.6)
$$

Example 8.2.2 If (8.2.6) is applied to

$$
A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 6 & 10 & 15 & 21 \\ 1 & 4 & 10 & 20 & 35 & 56 \\ 1 & 5 & 15 & 35 & 70 & 126 \\ 1 & 6 & 21 & 56 & 126 & 252 \end{bmatrix}
$$

with $x_0 = [1,\ 1,\ 1,\ 1,\ 1,\ 1]^T/6$, then

k	μ_k
0	153.8333
1	120.0571
2	49.5011
3	13.8687
4	15.4959
5	15.5534

The iteration is converging to the eigenvalue $\lambda = 15.5534732737$.

The Rayleigh quotient iteration almost always converges and when it does, the rate of convergence is cubic. We demonstrate this for the case $n = 2$. Without loss of generality, we may assume that $A = \text{diag}(\lambda_1, \lambda_2)$, with $\lambda_1 > \lambda_2$. Denoting x_k by

$$
x_k = \begin{bmatrix} c_k \\ s_k \end{bmatrix} \qquad c_k^2 + s_k^2 = 1
$$

it follows that $\mu_k = \lambda_1 c_k^2 + \lambda_1 s_k^2$ in (8.2.6) and

$$
z_{k+1} = \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} c_k / s_k^2 \\ -s_k / c_k^2 \end{bmatrix}.
$$

A calculation shows that

$$
c_{k+1} = \frac{c_k^3}{\sqrt{c_k^6 + s_k^6}} \qquad s_{k+1} = \frac{-s_k^3}{\sqrt{c_k^6 + s_k^6}}. \qquad (8.2.7)
$$

From these equations it is clear that the x_k converge cubically to either $\text{span}\{e_1\}$ or $\text{span}\{e_2\}$ provided $|c_k| \neq |s_k|$.

Details associated with the practical implementation of the Rayleigh quotient iteration may be found in Parlett (1974).

8.2.4 Orthogonal Iteration

A straightforward generalization of the power method can be used to compute higher-dimensional invariant subspaces. Let r be a chosen integer satisfying $1 \le r \le n$. Given an n-by-r matrix Q_0 with orthonormal columns, the method of *orthogonal iteration* generates a sequence of matrices $\{Q_k\} \subseteq \mathbb{R}^{n\times r}$ as follows:

$$\begin{array}{ll} \textbf{for } k = 1,2,\ldots & \\ \quad Z_k = AQ_{k-1} & \\ \quad Q_kR_k = Z_k & \text{(QR factorization)} \\ \textbf{end} & \end{array} \tag{8.2.8}$$

Note that if $r = 1$, then this is just the power method. Moreover, the sequence $\{Q_ke_1\}$ is precisely the sequence of vectors produced by the power iteration with starting vector $q^{(0)} = Q_0e_1$.

In order to analyze the behavior of (8.2.8), assume that

$$Q^TAQ = D = \text{diag}(\lambda_i) \qquad |\lambda_1| \ge |\lambda_2| \ge \cdots \ge |\lambda_n| \tag{8.2.9}$$

is a Schur decomposition of $A \in \mathbb{R}^{n\times n}$. Partition Q and D as follows:

$$Q = \begin{bmatrix} Q_\alpha & Q_\beta \end{bmatrix} \quad \begin{matrix} \\ r \quad n-r \end{matrix} \qquad D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} \quad \begin{matrix} \\ r \quad n-r \end{matrix} \tag{8.2.10}$$

If $|\lambda_r| > |\lambda_{r+1}|$, then

$$D_r(A) = \text{ran}(Q_\alpha)$$

is the *dominant* invariant subspace of dimension r. It is the unique invariant subspace associated with the eigenvalues $\lambda_1, \ldots, \lambda_r$.

The following theorem shows that with reasonable assumptions, the subspaces $\text{ran}(Q_k)$ generated by (8.2.8) converge to $D_r(A)$ at a rate proportional to $|\lambda_{r+1}/\lambda_r|^k$.

Theorem 8.2.2 *Let the Schur decomposition of $A \in \mathbb{R}^{n\times n}$ be given by (8.2.9) and (8.2.10) with $n \ge 2$. Assume that $|\lambda_r| > |\lambda_{r+1}|$ and that the n-by-r matrices $\{Q_k\}$ are defined by (8.2.8). If $\theta \in [0, \pi/2]$ is specified by*

$$\cos(\theta) = \min_{\substack{u \in D_r(A) \\ v \in \text{ran}(Q_0)}} \frac{|u^Tv|}{\| u \|_2 \| v \|_2} > 0\,,$$

then

$$\text{dist}(D_r(A), \text{ran}(Q_k)) \le \tan(\theta) \left| \frac{\lambda_{r+1}}{\lambda_r} \right|^k .$$

See also Theorem 7.3.1.

Proof. By induction it can be shown that

$$A^k Q_0 = Q_k \left(R_k \cdots R_1\right)$$

and so with the partitionings (8.2.10) we have

$$\begin{bmatrix} D_1^k & 0 \\ 0 & D_2^k \end{bmatrix} \begin{bmatrix} Q_\alpha^T Q_0 \\ Q_\beta^T Q_0 \end{bmatrix} = \begin{bmatrix} Q_\alpha^T Q_k \\ Q_\beta^T Q_k \end{bmatrix} \left(R_k \cdots R_1\right).$$

If

$$Q^T Q_k = \left[Q_\alpha \, , \, Q_\beta\right]^T Q_k = \begin{bmatrix} Q_\alpha^T Q_k \\ Q_\beta^T Q_k \end{bmatrix} \equiv \begin{bmatrix} V_k \\ W_k \end{bmatrix},$$

then

$$\begin{aligned} \cos(\theta_{min}) &= \sigma_r(V_0) = \sqrt{1 - \| W_0 \|_2^2} \\ \text{dist}(D_r(A), \text{ran}(Q_k)) &= \| W_k \|_2 \\ D_1^k V_0 &= V_k \left(R_k \cdots R_1\right) \\ D_2^k W_0 &= W_k \left(R_k \cdots R_1\right) \end{aligned}$$

It follows that V_0 is nonsingular which in turn implies that V_k and $(R_k \cdots R_1)$ are also nonsingular. Thus,

$$\begin{aligned} W_k &= D_2^k W_0 \left(R_k \cdots R_1\right)^{-1} = D_2^k W_0 \left(V_k^{-1} D_1^k V_0\right)^{-1} \\ &= D_2^k W_0 V_0^{-1} D_1^{-k} V_k \end{aligned}$$

and so

$$\begin{aligned} \| W_k \|_2 &\le \| D_2^k \|_2 \, \| W_0 \|_2 \, \| V_0^{-1} \|_2 \, \| D_1^{-k} \|_2 \, \| V_k \|_2 \\ &\le |\lambda_{r+1}|^k \, \sin(\theta) \, \frac{1}{\cos(\theta)} \, \frac{1}{|\lambda_r|^k} = \tan(\theta) \left| \frac{\lambda_{r+1}}{\lambda_r} \right|^k . \; \Box \end{aligned}$$

Example 8.2.3 If (8.2.8) is applied to the matrix of Example 8.2.1 with $r = 2$ and $Q_0 = I_4(:, 1{:}2)$, then

k	dist($D_2(A)$,ran(Q_k))
1	0.8806
2	0.4091
3	0.1121
4	0.0313
5	0.0106
6	0.0044
7	0.0020
8	0.0010
9	0.0005
10	0.0002

8.2.5 The QR Iteration

Consider what happens when we apply the method of orthogonal iteration (8.2.8) with $r = n$. Let $Q^TAQ = \text{diag}(\lambda_1, \ldots, \lambda_n)$ be the Schur decomposition and assume

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|.$$

If $Q = [\, q_1, \ldots, q_n \,]$ and $Q_k = \left[\, q_1^{(k)}, \ldots, q_n^{(k)} \,\right]$ and

$$\text{dist}(D_i(A), \text{span}\{q_1^{(0)}, \ldots, q_i^{(0)}\}) \;<\; 1 \qquad (8.2.11)$$

for $i = 1{:}n-1$, then it follows from Theorem 8.2.2 that

$$\text{dist}(\text{span}\{q_1^{(k)}, \ldots, q_i^{(k)}\}, \text{span}\{q_1, \ldots, q_i\}) \;=\; O\left(\left|\frac{\lambda_{i+1}}{\lambda_i}\right|^k\right).$$

for $i = 1{:}n-1$. This implies that the matrices T_k defined by

$$T_k \;=\; Q_k^T A Q_k$$

are converging to diagonal form. Thus, it can be said that the method of orthogonal iteration computes a Schur decomposition if $r = n$ and the original iterate $Q_0 \in \mathbb{R}^{n \times n}$ is not deficient in the sense of (8.2.11).

The QR iteration arises by considering how to compute the matrix T_k directly from its predecessor T_{k-1}. On the one hand, we have from (8.2.1) and the definition of T_{k-1} that

$$T_{k-1} = Q_{k-1}^T A Q_{k-1} = Q_{k-1}^T (A Q_{k-1}) = (Q_{k-1}^T Q_k) R_k.$$

On the other hand,

$$T_k = Q_k^T A Q_k = (Q_k^T A Q_{k-1})(Q_{k-1}^T Q_k) = R_k (Q_{k-1}^T Q_k).$$

Thus, T_k is determined by computing the QR factorization of T_{k-1} and then multiplying the factors together in reverse order. This is precisely what is done in (8.2.1).

Example 8.2.4 If the QR iteration (8.2.1) is applied to the matrix in Example 8.2.1, then after 10 iterations

$$T_{10} = \begin{bmatrix} 11.9907 & -0.1926 & -0.0004 & 0.0000 \\ -0.1926 & 8.0093 & -0.0029 & 0.0001 \\ -0.0004 & -0.0029 & -4.0000 & 0.0007 \\ 0.0000 & 0.0001 & 0.0007 & -2.0000 \end{bmatrix}.$$

The off-diagonal entries of the T_k matrices go to zero as follows:

k	$\|T_k(2,1)\|$	$\|T_k(3,1)\|$	$\|T_k(4,1)\|$	$\|T_k(3,2)\|$	$\|T_k(4,2)\|$	$\|T_k(4,3)\|$
1	3.9254	1.8122	3.3892	4.2492	2.8367	1.1679
2	2.6491	1.2841	2.1908	1.1587	3.1473	0.2294
3	2.0147	0.6154	0.5082	0.0997	0.9859	0.0748
4	1.6930	0.2408	0.0970	0.0723	0.2596	0.0440
5	1.2928	0.0866	0.0173	0.0665	0.0667	0.0233
6	0.9222	0.0299	0.0030	0.0405	0.0169	0.0118
7	0.6346	0.0101	0.0005	0.0219	0.0043	0.0059
8	0.4292	0.0034	0.0001	0.0113	0.0011	0.0030
9	0.2880	0.0011	0.0000	0.0057	0.0003	0.0015
10	0.1926	0.0004	0.0000	0.0029	0.0001	0.0007

Note that a single QR iteration involves $O(n^3)$ flops. Moreover, since convergence is only linear (when it exists), it is clear that the method is a prohibitively expensive way to compute Schur decompositions. Fortunately, these practical difficulties can be overcome as we show in the next section.

Problems

P8.2.1 Suppose $A_0 \in \mathbb{R}^{n\times n}$ is symmetric and positive definite and consider the following iteration:

```
for k = 1, 2, ...
    A_{k-1} = G_k G_k^T      (Cholesky)
    A_k = G_k^T G_k
end
```

(a) Show that this iteration is defined. (b) Show that if $A_0 = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ with $a \geq c$ has eigenvalues $\lambda_1 \geq \lambda_2 > 0$, then the A_k converge to $\text{diag}(\lambda_1, \lambda_2)$.

P8.2.2 Prove (8.2.7).

P8.2.3 Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and define the function $f: \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ by

$$f\begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{bmatrix} Ax - \lambda x \\ (x^Tx - 1)/2 \end{bmatrix}$$

where $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$. Suppose x_+ and λ_+ are produced by applying Newton's method to f at the "current point" defined by x_c and λ_c. Give expressions for x_+ and λ_+ assuming that $\| x_c \|_2 = 1$ and $\lambda_c = x_c^T A x_c$.

Notes and References for Sec. 8.2

The following references are concerned with the method of orthogonal iteration (a.k.a. the method of simultaneous iteration):

G.W. Stewart (1969). "Accelerating The Orthogonal Iteration for the Eigenvalues of a Hermitian Matrix," *Numer. Math.* *13*, 362–76.

M. Clint and A. Jennings (1970). "The Evaluation of Eigenvalues and Eigenvectors of Real Symmetric Matrices by Simultaneous Iteration," *Comp. J.* *13*, 76–80.

H. Rutishauser (1970). "Simultaneous Iteration Method for Symmetric Matrices," *Numer. Math.* *16*, 205–23. See also Wilkinson and Reinsch (1971,pp.284–302).

References for the Rayleigh quotient method include

J. Vandergraft (1971). "Generalized Rayleigh Methods with Applications to Finding Eigenvalues of Large Matrices," *Lin. Alg. and Its Applic.* *4*, 353–68.

B.N. Parlett (1974). "The Rayleigh Quotient Iteration and Some Generalizations for Nonnormal Matrices," *Math. Comp.* *28*, 679-93.

R.A. Tapia and D.L. Whitley (1988). "The Projected Newton Method Has Order $1+\sqrt{2}$ for the Symmetric Eigenvalue Problem," *SIAM J. Num. Anal.* *25*, 1376–1382.

S. Batterson and J. Smillie (1989). "The Dynamics of Rayleigh Quotient Iteration," *SIAM J. Num. Anal.* *26*, 624–636.

C. Beattie and D.W. Fox (1989). "Localization Criteria and Containment for Rayleigh Quotient Iteration," *SIAM J. Matrix Anal. Appl.* *10*, 80–93.

P.T.P. Tang (1994). "Dynamic Condition Estimation and Rayleigh-Ritz Approximation," *SIAM J. Matrix Anal. Appl.* *15*, 331-346.

8.3 The Symmetric QR Algorithm

The symmetric QR iteration (8.2.1) can be made very efficient in two ways. First, we show how to compute an orthogonal U_0 such that $U_0^T A U = T$ is tridiagonal. With this reduction, the iterates produced by (8.2.1) are all tridiagonal and this reduces the work per step to $O(n^2)$. Second, the idea of shifts are introduced and with this change the convergence to diagonal form proceeds at a cubic rate. This is far better than having the off-diagonal entries going to to zero like $|\lambda_{i+1}/\lambda_i|^k$ as discussed in §8.2.5.

8.3.1 Reduction to Tridiagonal Form

If A is symmetric, then it is possible to find an orthogonal Q such that

$$Q^T A Q = T \qquad (8.3.1)$$

is tridiagonal. We call this the *tridiagonal decomposition* and as a compression of data, it represents a very big step towards diagonalization.

We show how to compute (8.3.1) with Householder matrices. Suppose that Householder matrices $P_1, \ldots, P_{k-1}$ have been determined such that if $A_{k-1} = (P_1 \cdots P_{k-1})^T A (P_1 \cdots P_{k-1})$, then

$$A_{k-1} = \begin{bmatrix} B_{11} & B_{12} & 0 \\ B_{21} & B_{22} & B_{23} \\ 0 & B_{32} & B_{33} \end{bmatrix} \begin{matrix} k-1 \\ 1 \\ n-k \end{matrix}$$
$$\qquad\quad k-1 \quad\ \ 1 \quad\ \ n-k$$

is tridiagonal through its first $k-1$ columns. If $\tilde{P}_k$ is an order $n-k$ Householder matrix such that $\tilde{P}_k B_{32}$ is a multiple of $I_{n-k}(:,1)$ and if $P_k =$

$\text{diag}(I_k, \bar{P}_k)$, then the leading k-by-k principal submatrix of

$$A_k = P_k A_{k-1} P_k = \begin{bmatrix} B_{11} & B_{12} & 0 \\ B_{21} & B_{22} & B_{23}\bar{P}_k \\ 0 & \bar{P}_k B_{32} & \bar{P}_k B_{33} \bar{P}_k \end{bmatrix} \begin{matrix} k-1 \\ 1 \\ n-k \end{matrix}$$
$$\qquad\qquad k-1 \qquad 1 \qquad n-k$$

is tridiagonal. Clearly, if $U_0 = P_1 \cdots P_{n-2}$, then $U_0^T A U_0 = T$ is tridiagonal.

In the calculation of A_k it is important to exploit symmetry during the formation of the matrix $\bar{P}_k B_{33} \bar{P}_k$. To be specific, suppose that $\bar{P}_k$ has the form

$$\bar{P}_k = I - \beta v v^T \qquad \beta = 2/v^T v, \quad 0 \neq v \in \mathbb{R}^{n-k}.$$

Note that if $p = \beta B_{33} v$ and $w = p - (\beta p^T v/2)v$, then

$$\bar{P}_k B_{33} \bar{P}_k = B_{33} - v w^T - w v^T.$$

Since only the upper triangular portion of this matrix needs to be calculated, we see that the transition from A_{k-1} to A_k can be accomplished in only $4(n-k)^2$ flops.

Algorithm 8.3.1 (Householder Tridiagonalization) Given a symmetric $A \in \mathbb{R}^{n \times n}$, the following algorithm overwrites A with $T = Q^T A Q$, where T is tridiagonal and $Q = H_1 \cdots H_{n-2}$ is the product of Householder transformations.

```
for k = 1:n − 2
    [v, β] = house(A(k + 1:n, k))
    p = βA(k + 1:n, k + 1:n)v
    w = p − (βp^T v/2)v
    A(k + 1, k) = || A(k + 1:n, k) ||_2; A(k, k + 1) = A(k + 1.k)
    A(k + 1:n, k + 1:n) = A(k + 1:n, k + 1:n) − vw^T − wv^T
end
```

This algorithm requires $4n^3/3$ flops when symmetry is exploited in calculating the rank-2 update. The matrix Q can be stored in factored form in the subdiagonal portion of A. If Q is explicitly required, then it can be formed with an additional $4n^3/3$ flops.

Example 8.3.1

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & .6 & .8 \\ 0 & .8 & -.6 \end{bmatrix}^T \begin{bmatrix} 1 & 3 & 4 \\ 3 & 2 & 8 \\ 4 & 8 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & .6 & .8 \\ 0 & .8 & -.6 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 0 \\ 5 & 10.32 & 1.76 \\ 0 & 1.76 & -5.32 \end{bmatrix}.$$

Note that if T has a zero subdiagonal, then the eigenproblem splits into a pair of smaller eigenproblems. In particular, if $t_{k+1,k} = 0$, then $\lambda(T) =$

$\lambda(T(1{:}k, 1{:}k)) \cup \lambda(T(k+1{:}n, k+1{:}n))$. If T has no zero subdiagonal entries, then it is said to be *unreduced.*

Let $\hat{T}$ denote the computed version of T obtained by Algorithm 8.3.1. It can be shown that $\hat{T} = \tilde{Q}^T(A + E)\tilde{Q}$ where $\tilde{Q}$ is exactly orthogonal and E is a symmetric matrix satisfying $\| E \|_F \le c\mathbf{u}\| A \|_F$ where c is a small constant. See Wilkinson (1965, p. 297).

8.3.2 Properties of the Tridiagonal Decomposition

We prove two theorems about the tridiagonal decomposition both of which have key roles to play in the sequel. The first connects (8.3.1) to the QR factorization of a certain *Krylov matrix.* These matrices have the form

$$K(A, v, k) = \left[v,\ Av, \cdots,\ A^{k-1}v\right] \qquad A \in \mathbb{R}^{n \times n},\ v \in \mathbb{R}^n.$$

Theorem 8.3.1 *If $Q^TAQ = T$ is the tridiagonal decomposition of the symmetric matrix $A \in \mathbb{R}^{n\times n}$, then $Q^TK(A, Q(:,1), n) = R$ is upper triangular. If R is nonsingular, then T is unreduced. If R is singular and k is the smallest index so $r_{kk} = 0$, then k is also the smallest index so $t_{k,k-1}$ is zero. See also Theorem 7.4.3.*

Proof. It is clear that if $q_1 = Q(:,1)$, then

$$\begin{aligned} Q^TK(A, Q(:,1), n) &= \left[\, Q^Tq_1,\ (Q^TAQ)(Q^Tq_1), \ldots, (Q^TAQ)^{n-1}(Q^Tq_1) \,\right] \\ &= \left[\, e_1,\ Te_1, \ldots, T^{n-1}e_1 \,\right] \;=\; R \end{aligned}$$

is upper triangular with the property that $r_{11} = 1$ and $r_{ii} = t_{21}t_{32}\cdots t_{i,i-1}$ for $i = 2{:}n$. Clearly, if R is nonsingular, then T is unreduced. If R is singular and r_{kk} is its first zero diagonal entry, then $k \ge 2$ and $t_{k,k-1}$ is the first zero subdiagonal entry. □

The next result shows that Q is essentially unique once $Q(:,1)$ is specified.

Theorem 8.3.2 (Implicit Q Theorem) *Suppose $Q = [\, q_1, \ldots, q_n \,]$ and $V = [\, v_1, \ldots, v_n \,]$ are orthogonal matrices with the property that both $Q^TAQ = T$ and $V^TAV = S$ are tridiagonal where $A \in \mathbb{R}^{n\times n}$ is symmetric. Let k denote the smallest positive integer for which $t_{k+1,k} = 0$, with the convention that $k = n$ if T is unreduced. If $v_1 = q_1$, then $v_i = \pm q_i$ and $|t_{i,i-1}| = |s_{i,i-1}|$ for $i = 2{:}k$. Moreover, if $k < n$, then $s_{k+1,k} = 0$. See also Theorem 7.4.2.*

Proof. Define the orthogonal matrix $W = Q^TV$ and observe that $W(:,1) = I_n(:,1) = e_1$ and $W^TTW = S$. By Theorem 8.3.1, $W^TK(T, e_1, k)$ is upper triangular with full column rank. But $K(T, e_1, k)$ is upper triangular and so by the essential uniqueness of the thin QR factorization,

$$W(:, 1{:}k) = I_n(:, 1{:}k)\mathrm{diag}(\pm 1, \ldots, \pm 1).$$

This says that $Q(:,i) = \pm V(:,i)$ for $i = 1:k$. The comments about the subdiagonal entries follows from this since $t_{i+1,i} = Q(:,i+1)^T AQ(:,i)$ and $s_{i+1,i} = V(:,i+1)^T AV(:,i)$ for $i = 1:n-1$. □

8.3.3 The QR Iteration and Tridiagonal Matrices

We quickly state four facts that pertain to the QR iteration and tridiagonal matrices. Complete verifications are straight forward.

1. *Preservation of Form.* If $T = QR$ is the QR factorization of a symmetric tridiagonal matrix $T \in \mathbb{R}^{n\times n}$, then Q has lower bandwidth 1 and R has upper bandwidth 2 and it follows that

$$T_+ = RQ = Q^T(QR)Q = Q^TTQ$$

is also symmetric and tridiagonal.

2. *Shifts.* If $s \in \mathbb{R}$ and $T - sI = QR$ is the QR factorization, then

$$T_+ = RQ + sI = Q^TTQ$$

is also tridiagonal. This is called a *shifted* QR step.

3. *Perfect Shifts.* If T is unreduced, then the first $n-1$ columns of $T-sI$ are independent regardless of s. Thus, if $s \in \lambda(T)$ and

$$QR = T - sI$$

is a QR factorization, then $r_{nn} = 0$ and the last column of $T_+ = RQ + sI$ equals $sI_n(:,n) = se_n$.

4. *Cost.* If $T \in \mathbb{R}^{n\times n}$ is tridiagonal, then its QR factorization can be computed by applying a sequence of $n-1$ Givens rotations:

for $k = 1:n-1$
  $[c,s] = \textbf{givens}(t_{kk}, t_{k+1,k})$
  $m = \min\{k+2, n\}$
  $T(k:k+1, k:m) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T T(k:k+1, k:m)$
end

This requires $O(n)$ flops. If the rotations are accumulated, then $O(n^2)$ flops are needed.

8.3.4 Explicit Single Shift QR Iteration

If s is a good approximate eigenvalue, then we suspect that the $(n, n-1)$ will be small after a QR step with shift s. This is the philosophy behind the following iteration:

$$
\begin{aligned}
& T = U_0^T A U_0 \qquad \text{(tridiagonal)} \\
& \textbf{for } k = 0, 1, \ldots \\
& \qquad \text{Determine real shift } \mu. \\
& \qquad T - \mu I = UR \qquad \text{(QR factorization)} \\
& \qquad T = RU + \mu I \\
& \textbf{end}
\end{aligned}
\tag{8.3.2}
$$

If

$$
T = \begin{bmatrix} a_1 & b_1 & & \cdots & 0 \\ b_1 & a_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & b_{n-1} \\ 0 & \cdots & & b_{n-1} & a_n \end{bmatrix}.
$$

then one reasonable choice for the shift is $\mu = a_n$. However, a more effective choice is to shift by the eigenvalue of

$$
T(n-1{:}n, n-1{:}n) = \begin{bmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{bmatrix}
$$

that is closer to a_n. This is known as the *Wilkinson shift* and it is given by

$$
\mu = a_n + d - \text{sign}(d)\sqrt{d^2 + b_{n-1}^2} \tag{8.3.3}
$$

where $d = (a_{n-1} - a_n)/2$. Wilkinson (1968b) has shown that (8.3.2) is cubically convergent with either shift strategy, but gives heuristic reasons why (8.3.3) is preferred.

8.3.5 Implicit Shift Version

It is possible to execute the transition from T to $T_+ = RU + \mu I = U^T T U$ without explicitly forming the matrix $T - \mu I$. This has advantages when the shift is much larger than some of the a_i. Let $c = \cos(\theta)$ and $s = \sin(\theta)$ be computed such that

$$
\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_1 - \mu \\ b_1 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}.
$$

If we set $G_1 = G(1,2,\theta)$ then $G_1e_1 = Ue_1$ and

$$T \leftarrow G_1^TTG_1 = \begin{bmatrix} \times & \times & + & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ + & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

We are thus in a position to apply the Implicit Q theorem provided we can compute rotations $G_2, \ldots, G_{n-1}$ with the property that if $Z = G_1G_2 \cdots G_{n-1}$ then $Ze_1 = G_1e_1 = Ue_1$ *and* Z^TTZ is tridiagonal.

Note that the first column of Z and U are identical provided we take each G_i to be of the form $G_i = G(i, i+1, \theta_i)$, $i = 2{:}n-1$. But G_i of this form can be used to chase the unwanted nonzero element "+" out of the matrix $G_1^TTG_1$ as follows:

$$\xrightarrow{G_2} \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & + & 0 & 0 \\ 0 & \times & \times & \times & 0 & 0 \\ 0 & + & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix} \quad \xrightarrow{G_3} \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & + & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & + & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

$$\xrightarrow{G_4} \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & + \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & + & \times & \times \end{bmatrix} \quad \xrightarrow{G_5} \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Thus, it follows from the Implicit Q theorem that the tridiagonal matrix Z^TTZ produced by this zero-chasing technique is essentially the same as the tridiagonal matrix T obtained by the explicit method. (We may assume that all tridiagonal matrices in question are unreduced for otherwise the problem decouples.)

Note that at any stage of the zero-chasing, there is only one nonzero entry outside the tridiagonal band. How this nonzero entry moves down the matrix during the update $T \leftarrow G_k^TTG_k$ is illustrated in the following:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} a_k & b_k & z_k & 0 \\ b_k & a_p & b_p & 0 \\ z_k & b_p & a_q & b_q \\ 0 & 0 & b_q & a_r \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_k & b_k & 0 & 0 \\ b_k & a_p & b_p & z_p \\ 0 & b_p & a_q & b_q \\ 0 & z_p & b_q & a_r \end{bmatrix}.$$

Here $(p,q,r) = (k+1, k+2, k+3)$. This update can be performed in about

26 flops once c and s have been determined from the equation $b_k s + z_k c = 0$. Overall we obtain

Algorithm 8.3.2 (Implicit Symmetric QR Step with Wilkinson Shift) Given an unreduced symmetric tridiagonal matrix $T \in \mathbb{R}^{n\times n}$, the following algorithm overwrites T with $Z^T T Z$, where $Z = G_1 \cdots G_{n-1}$ is a product of Givens rotations with the property that $Z^T(T - \mu I)$ is upper triangular and μ is that eigenvalue of T's trailing 2-by-2 principal submatrix closer to t_{nn}.

$$
\begin{array}{l}
d = (t_{n-1,n-1} - t_{nn})/2 \\
\mu = t_{nn} - t_{n,n-1}^2 / \left(d + \text{sign}(d)\sqrt{d^2 + t_{n,n-1}^2} \right) \\
x = t_{11} - \mu \\
z = t_{21} \\
\textbf{for } k = 1{:}n-1 \\
\quad [\, c, s \,] = \textbf{givens}(x, z) \\
\quad T = G_k^T T G_k, \quad \text{where } G_k = G(k, k+1, \theta) \\
\quad \textbf{if } k < n-1 \\
\quad\quad x = t_{k+1,k} \\
\quad\quad z = t_{k+2,k} \\
\quad \textbf{end} \\
\textbf{end}
\end{array}
$$

This algorithm requires about $30n$ flops and n square roots. If a given orthogonal matrix Q is overwritten with $QG_1 \cdots G_{n-1}$, then an additional $6n^2$ flops are needed. Of course, in any practical implementation the tridiagonal matrix T would be stored in a pair of n-vectors and not in an n-by-n array.

Example 8.3.2 If Algorithm 8.3.2 is applied to

$$
T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 3 & .01 \\ 0 & 0 & .01 & 4 \end{bmatrix},
$$

then the new tridiagonal matrix T is given by

$$
T = \begin{bmatrix} .5000 & .5916 & 0 & 0 \\ .5916 & 1.785 & .1808 & 0 \\ 0 & .1808 & 3.7140 & .0000044 \\ 0 & 0 & .0000044 & 4.002497 \end{bmatrix}.
$$

Algorithm 8.3.2 is the basis of the symmetric QR algorithm—the standard means for computing the Schur decomposition of a dense symmetric matrix.

Algorithm 8.3.3 (Symmetric QR Algorithm) Given $A \in \mathbb{R}^{n\times n}$ (symmetric) and a tolerance *tol* greater than the unit roundoff, this algorithm computes an approximate symmetric Schur decomposition $Q^TAQ = D$. A is overwritten with the tridiagonal decomposition.

Use Algorithm 8.3.1, compute the tridiagonalization
    $T = (P_1 \cdots P_{n-2})^T A (P_1 \cdots P_{n-2})$.
    Set $D = T$ and if Q is desired, form $Q = P_1 \cdots P_{n-2}$. See §5.1.6.
until $q = n$
    For $i = 1{:}n-1$, set $d_{i+1,i}$ and $d_{i,i+1}$ to zero if
        $|d_{i+1,i}| = |d_{i,i+1}| \le tol(|d_{ii}| + |d_{i+1,i+1}|)$
    Find the largest q and the smallest p such that if

$$D = \begin{bmatrix} D_{11} & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & D_{33} \end{bmatrix} \begin{matrix} p \\ n-p-q \\ q \end{matrix}$$
$$\qquad\quad p \qquad n-p-q \qquad q$$

    then D_{33} is diagonal and D_{22} is unreduced.
    if $q < n$
        Apply Algorithm 8.3.2 to D_{22}:
            $D = \text{diag}(I_p, \bar{Z}, I_q)^T D \ \text{diag}(I_p, \bar{Z}, I_q)$
        If Q is desired, then $Q = Q \ \text{diag}(I_p, \bar{Z}, I_q)$.
    end
end

This algorithm requires about $4n^3/3$ flops if Q is not accumulated and about $9n^3$ flops if Q is accumulated.

Example 8.3.3 Suppose Algorithm 8.3.3 is applied to the tridiagonal matrix

$$A = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 2 & 3 & 4 & 0 \\ 0 & 4 & 5 & 6 \\ 0 & 0 & 6 & 7 \end{bmatrix}$$

The subdiagonal entries change as follows during the execution of Algorithm 8.3.3:

Iteration	a_{21}	a_{32}	a_{43}
1	1.6817	3.2344	.8649
2	1.6142	2.5755	.0006
3	1.6245	1.6965	10^{-13}
4	1.6245	1.6965	converg.
5	1.5117	.0150	
6	1.1195	10^{-9}	
7	.7071	converg.	
8	converg.		

Upon completion we find $\lambda(A) = \{-2.4848, .7046, 4.9366, 12.831\}$.

The computed eigenvalues $\hat{\lambda}_i$ obtained via Algorithm 8.3.3 are the exact eigenvalues of a matrix that is near to A, i.e., $Q_0^T(A+E)Q_0 = \text{diag}(\hat{\lambda}_i)$ where $Q_0^TQ_0 = I$ and $\| E \|_2 \approx \mathbf{u}\| A \|_2$. Using Corollary 8.1.6 we know that the absolute error in each $\hat{\lambda}_i$ is small in the sense that $|\hat{\lambda}_i - \lambda_i| \approx \mathbf{u}\| A \|_2$. If $\hat{Q} = [\, \hat{q}_1, \ldots, \hat{q}_n \,]$ is the computed matrix of orthonormal eigenvectors, then the accuracy of $\hat{q}_i$ depends on the separation of λ_i from the remainder of the spectrum. See Theorem 8.1.12.

If all of the eigenvalues and a few of the eigenvectors are desired, then it is cheaper not to accumulate Q in Algorithm 8.3.3. Instead, the desired eigenvectors can be found via inverse iteration with T. See §8.2.2. Usually just one step is sufficient to get a good eigenvector, even with a random initial vector.

If just a few eigenvalues and eigenvectors are required, then the special techniques in §8.5 are appropriate.

It is interesting to note the connection between Rayleigh quotient iteration and the symmetric QR algorithm. Suppose we apply the latter to the tridiagonal matrix $T \in \mathbb{R}^{n\times n}$ with shift $\sigma = e_n^TTe_n = t_{nn}$ where $e_n = I_n(:,n)$. If $T - \sigma I = QR$, then we obtain $T = RQ + \sigma I$. From the equation $(T - \sigma I)Q = R^T$ it follows that

$$(T - \sigma I)q_n = r_{nn}e_n,$$

where q_n is the last column of the orthogonal matrix Q. Thus, if we apply (8.2.6) with $x_0 = e_n$, then $x_1 = q_n$.

8.3.6 Orthogonal Iteration with Ritz Acceleration

Recall from §8.2.4 that an orthogonal iteration step involves a matrix-matrix product and a QR factorization:

$$\begin{aligned} Z_k &= A\tilde{Q}_{k-1} \\ \tilde{Q}_kR_k &= Z_k \quad \text{(QR factorization)} \end{aligned}$$

Theorem 8.1.14 says that we can minimize $\| A\tilde{Q}_k - \tilde{Q}_kS \|_F$ by setting $S = S_k \equiv \tilde{Q}_k^TA\tilde{Q}_k$. If $U_k^TS_kU_k = D_k$ is the Schur decomposition of $S_k \in \mathbb{R}^{r\times r}$ and $Q_k = \tilde{Q}_kU_k$, then

$$\| AQ_k - Q_kD_k \|_F = \| A\tilde{Q}_k - \tilde{Q}_kS_k \|_F$$

showing that the columns of Q_k are the best possible basis to take after k steps from the standpoint of minimizing the residual. This defines the *Ritz acceleration* idea:

$$
\begin{array}{l}
Q_0 \in \mathbb{R}^{n\times p} \text{ given with } Q_0^TQ_0 = I_p \\
\textbf{for } k = 1,2,\ldots \\
\quad Z_k = AQ_{k-1} \\
\quad \tilde{Q}_kR_k = Z_k \quad \text{(QR factorization)} \\
\quad S_k = \tilde{Q}_k^TA\tilde{Q}_k \\
\quad U_k^TS_kU_k = D_k \quad \text{(Schur decomposition)} \\
\quad Q_k = \tilde{Q}_kU_k \\
\textbf{end}
\end{array}
\tag{8.3.6}
$$

It can be shown that if

$$D_k = \operatorname{diag}(\theta_1^{(k)},\ldots,\theta_r^{(k)}) \qquad |\theta_1^{(k)}| \geq \cdots \geq |\theta_r^{(k)}|$$

then

$$|\theta_i^{(k)} - \lambda_i(A)| = O\left(\left|\frac{\lambda_{r+1}}{\lambda_i}\right|^k\right) \qquad i = 1{:}r$$

Recall that Theorem 8.2.2 says the eigenvalues of $\tilde{Q}_k^TA\tilde{Q}_k$ converge with rate $|\lambda_{r+1}/\lambda_r|^k$. Thus, the Ritz values converge at a more favorable rate. For details, see Stewart (1969).

Example 8.3.4 If we apply (8.3.6) with

$$A = \begin{bmatrix} 100 & 1 & 1 & 1 \\ 1 & 99 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad Q_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

then

k	$\text{dist}\{D_2(A), Q_k\}$
0	$.2 \times 10^{-1}$
1	$.5 \times 10^{-3}$
2	$.1 \times 10^{-4}$
3	$.3 \times 10^{-6}$
4	$.8 \times 10^{-8}$

Clearly, convergence is taking place at the rate $(2/99)^k$.

Problems

P8.3.1 Suppose λ is an eigenvalue of a symmetric tridiagonal matrix T. Show that if λ has algebraic multiplicity k, then at least $k-1$ of T's subdiagonal elements are zero.

P8.3.2 Suppose A is symmetric and has bandwidth p. Show that if we perform the shifted QR step $A - \mu I = QR$, $A = RQ + \mu I$, then A has bandwidth p.

P8.3.3 Suppose $B \in \mathbb{R}^{n\times n}$ is upper bidiagonal with diagonal entries $d(1{:}n)$ and superdiagonal entries $f(1{:}n-1)$. State and prove a singular value version of Theorem 8.3.1.

P8.3.4 Let $A = \begin{bmatrix} w & x \\ x & z \end{bmatrix}$ be real and suppose we perform the following shifted QR step: $A - zI = UR$, $\bar{A} = RU + zI$. Show that if $\bar{A} = \begin{bmatrix} \bar{w} & \bar{x} \\ \bar{x} & \bar{z} \end{bmatrix}$ then

$$\bar{w} = w + x^2(w - z)/[(w - z)^2 + x^2]$$
$$\bar{z} = z - x^2(w - z)/[(w - z)^2 + x^2]$$
$$\bar{x} = -x^3/[(w - z)^2 + x^2].$$

P8.3.5 Suppose $A \in \mathbb{C}^{n\times n}$ is Hermitian. Show how to construct unitary Q such that $Q^H AQ = T$ is real, symmetric, and tridiagonal.

P8.3.6 Show that if $A = B + iC$ is Hermitian, then $M = \begin{bmatrix} B & -C \\ C & B \end{bmatrix}$ is symmetric. Relate the eigenvalues and eigenvectors of A and M.

P8.3.7 Rewrite Algorithm 8.2.2 for the case when A is stored in two n-vectors. Justify the given flop count.

P8.3.8 Suppose $A = S + \sigma uu^T$ where $S \in \mathbb{R}^{n\times n}$ is skew-symmetric ($A^T = -A$, $u \in \mathbb{R}^n$ has unit 2-norm, and $\sigma \in \mathbb{R}$. Show how to compute an orthogonal Q such that $Q^T AQ$ is tridiagonal and $Q^T u = I_n(:,1) = e_1$.

Notes and References for Sec. 8.3

The tridiagonalization of a symmetric matrix is discussed in

R.S. Martin and J.H. Wilkinson (1968). "Householder's Tridiagonalization of a Symmetric Matrix," *Numer. Math. 11,* 181-95. See also Wilkinson and Reinsch (1971, pp.212–26).
H.R. Schwartz (1968). "Tridiagonalization of a Symmetric Band Matrix," *Numer. Math. 12,* 231–41. See also Wilkinson and Reinsch (1971, pp.273–83).
N.E. Gibbs and W.G. Poole, Jr. (1974). "Tridiagonalization by Permutations," *Comm. ACM 17,* 20–24.

The first two references contain Algol programs. Algol procedures for the explicit and implicit tridiagonal QR algorithm are given in

H. Bowdler, R.S. Martin, C. Reinsch, and J.H. Wilkinson (1968). "The QR and QL Algorithms for Symmetric Matrices," *Numer. Math. 11,* 293–306. See also Wilkinson and Reinsch (1971, pp.227–40).
A. Dubrulle, R.S. Martin, and J.H. Wilkinson (1968). "The Implicit QL Algorithm," *Numer. Math. 12,* 377-83. see also Wilkinson and Reinsch (1971, pp.241 48).

The "QL" algorithm is identical to the QR algorithm except that at each step the matrix $T - \lambda I$ is factored into a product of an orthogonal matrix and a lower triangular matrix. Other papers concerned with these methods include

G.W. Stewart (1970). "Incorporating Original Shifts into the QR Algorithm for Symmetric Tridiagonal Matrices," *Comm. ACM 13,* 365–67.
A. Dubrulle (1970). "A Short Note on the Implicit QL Algorithm for Symmetric Tridiagonal Matrices," *Numer. Math. 15,* 450.

Extensions to Hermitian and skew-symmetric matrices are described in

D. Mueller (1966). "Householder's Method for Complex Matrices and Hermitian Matrices," *Numer. Math. 8,* 72–92.
R.C. Ward and L.J. Gray (1978). "Eigensystem Computation for Skew-Symmetric and A Class of Symmetric Matrices," *ACM Trans. Math. Soft. 4,* 278–85.

The convergence properties of Algorithm 8.2.3 are detailed in Lawson and Hanson (1974, Appendix B), as well as in

J.H. Wilkinson (1968b). "Global Convergence of Tridiagonal QR Algorithm With Origin Shifts," *Lin. Alg. and Its Applic. 1*, 409–20.
T.J. Dekker and J.F. Traub (1971). "The Shifted QR Algorithm for Hermitian Matrices," *Lin. Alg. and Its Applic. 4*, 137–54.
W. Hoffman and B.N. Parlett (1978). "A New Proof of Global Convergence for the Tridiagonal QL Algorithm," *SIAM J. Num. Anal. 15*, 929–37.
S. Batterson (1994). "Convergence of the Francis Shifted QR Algorithm on Normal Matrices," *Lin. Alg. and Its Applic. 207*, 181–195.

For an analysis of the method when it is applied to normal matrices see

C.P. Huang (1981). "On the Convergence of the QR Algorithm with Origin Shifts for Normal Matrices," *IMA J. Num. Anal. 1*, 127–33.

Interesting papers concerned with shifting in the tridiagonal QR algorithm include

F.L. Bauer and C. Reinsch (1968). "Rational QR Transformations with Newton Shift for Symmetric Tridiagonal Matrices," *Numer. Math. 11*, 264–72. See also Wilkinson and Reinsch (1971, pp.257–65).
G.W. Stewart (1970). "Incorporating Origin Shifts into the QR Algorithm for Symmetric Tridiagonal Matrices," *Comm. Assoc. Comp. Mach. 13*, 365–67.

Some parallel computation possibilities for the algorithms in this section are discussed in

S. Lo, B. Philippe, and A. Sameh (1987). "A Multiprocessor Algorithm for the Symmetric Tridiagonal Eigenvalue Problem," *SIAM J. Sci. and Stat. Comp. 8*, s155-s165.
H.Y. Chang and M. Salama (1988). "A Parallel Householder Tridiagonalization Strategy Using Scattered Square Decomposition," *Parallel Computing 6*, 297-312.

Another way to compute a specified subset of eigenvalues is via the rational QR algorithm. In this method, the shift is determined using Newton's method. This makes it possible to "steer" the iteration towards desired eigenvalues. See

C. Reinsch and F.L. Bauer (1968). "Rational QR Transformation with Newton's Shift for Symmetric Tridiagonal Matrices," *Numer. Math. 11*, 264–72. See also Wilkinson and Reinsch (1971, pp.257–65).

Papers concerned with the symmetric QR algorithm for banded matrices include

R.S. Martin and J.H. Wilkinson (1967). "Solution of Symmetric and Unsymmetric Band Equations and the Calculation of Eigenvectors of Band Matrices," *Numer. Math. 9*, 279–301. See also See also Wilkinson and Reinsch (1971, pp.70–92).
R.S. Martin, C. Reinsch, and J.H. Wilkinson (1970). "The QR Algorithm for Band Symmetric Matrices," *Numer. Math. 16*, 85–92. See also Wilkinson and Reinsch (1971, pp.266–72).

8.4 Jacobi Methods

Jacobi methods for the symmetric eigenvalue problem attract current attention because they are inherently parallel. They work by performing a sequence of orthogonal similarity updates $A \leftarrow Q^T A Q$ with the property that each new A, although full, is "more diagonal" than its predecessor. Eventually, the off-diagonal entries are small enough to be declared zero.

After surveying the basic ideas behind the Jacobi approach we develop a parallel Jacobi procedure.

8.4.1 The Jacobi Idea

The idea behind Jacobi's method is to systematically reduce the quantity

$$\text{off}(A) = \sqrt{\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}^2}\,,$$

i.e., the"norm" of the off-diagonal elements. The tools for doing this are rotations of the form

$$J(p,q,\theta) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \begin{matrix} \\ \\ p \\ \\ q \\ \\ \\ \end{matrix}$$

$$\qquad\qquad\qquad\qquad p \qquad\quad q$$

which we call *Jacobi rotations*. Jacobi rotations are no different from Givens rotations, c.f. §5.1.8. We submit to the name change in this section to honor the inventor.

The basic step in a Jacobi eigenvalue procedure involves (1) choosing an index pair (p,q) that satisfies $1 \le p < q \le n$, (2) computing a cosine-sine pair (c,s) such that

$$\begin{bmatrix} b_{pp} & b_{pq} \\ b_{qp} & b_{qq} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \tag{8.4.1}$$

is diagonal, and (3) overwriting A with $B = J^T A J$ where $J = J(p,q,\theta)$. Observe that the matrix B agrees with A except in rows and columns p

and q. Moreover, since the Frobenius norm is preserved by orthogonal transformations we find that

$$a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2 \;=\; b_{pp}^2 + b_{qq}^2 + 2b_{pq}^2 \;=\; b_{pp}^2 + b_{qq}^2$$

and so

$$\begin{aligned} \text{off}(B)^2 &= \| B \|_F^2 - \sum_{i=1}^{n} b_{ii}^2 \\ &= \| A \|_F^2 - \sum_{i=1}^{n} a_{ii}^2 \;+\; (a_{pp}^2 + a_{qq}^2 - b_{pp}^2 - b_{qq}^2) \\ &= \text{off}(A)^2 - 2a_{pq}^2 \,. \end{aligned} \qquad (8.4.2)$$

It is in this sense that A moves closer to diagonal form with each Jacobi step.

Before we discuss how the index pair (p, q) can be chosen, let us look at the actual computations associated with the (p, q) subproblem.

8.4.2 The 2-by-2 Symmetric Schur Decomposition

To say that we diagonalize in (8.4.1) is to say that

$$0 \;=\; b_{pq} \;=\; a_{pq}(c^2 - s^2) + (a_{pp} - a_{qq})cs. \qquad (8.4.3)$$

If $a_{pq} = 0$, then we just set $(c, s) = (1,0)$. Otherwise define

$$\tau \;=\; \frac{a_{qq} - a_{pp}}{2a_{pq}} \quad \text{and} \quad t \;=\; s/c$$

and conclude from (8.4.3) that $t = \tan(\theta)$ solves the quadratic

$$t^2 + 2\tau t - 1 = 0\,.$$

It turns out to be important to select the smaller of the two roots,

$$t \;=\; -\tau \pm \sqrt{1 + \tau^2}$$

whereupon c and s can be resolved from the formulae

$$c = 1/\sqrt{1 + t^2} \qquad s = tc\,.$$

Choosing t to be the smaller of the two roots ensures that $|\theta| \leq \pi/4$ and has the effect of minimizing the difference between B and A because

$$\| B - A \|_F^2 \;=\; 4(1 - c) \sum_{\substack{i=1 \\ i \neq p,q}}^{n} (a_{ip}^2 + a_{iq}^2) \;+\; 2a_{pq}^2/c^2$$

We summarize the 2-by-2 computations as follows:

Algorithm 8.4.1 Given an n-by-n symmetric A and integers p and q that satisfy $1 \leq p < q \leq n$, this algorithm computes a cosine-sine pair $(c\,,\,s)$ such that if $B = J(p,q,\theta)^T AJ(p,q,\theta)$ then $b_{pq} = b_{qp} = 0$.

```
function: [c , s] = sym.schur2(A, p, q)
    if A(p, q) ≠ 0
        τ = (A(q, q) − A(p, p))/(2A(p, q))
        if τ ≥ 0
            t = 1/(τ + √(1 + τ²));
        else
            t = −1/(−τ + √(1 + τ²));
        end
        c = 1/√(1 + t²)
        s = tc
    else
        c = 1
        s = 0
    end
```

8.4.3 The Classical Jacobi Algorithm

As we mentioned above, only rows and columns p and q are altered when the (p,q) subproblem is solved. Once **sym.schur2** determines the 2-by-2 rotation, then the update $A \leftarrow J(p,q,\theta)^T AJ(p,q,\theta)$ can be implemented in $6n$ flops if symmetry is exploited.

How do we choose the indices p and q? From the standpoint of maximizing the reduction of off(A) in (8.4.2), it makes sense to choose (p,q) so that a_{pq}^2 is maximal. This is the basis of the *classical* Jacobi algorithm.

Algorithm 8.4.2 (Classical Jacobi) Given a symmetric $A \in \mathbb{R}^{n\times n}$ and a tolerance $tol > 0$, this algorithm overwrites A with V^TAV where V is orthogonal and off$(V^TAV) \leq tol\| A \|_F$.

```
V = I_n; eps = tol‖ A ‖_F
while off(A) > eps
    Choose (p, q) so |a_pq| = max_{i≠j} |a_ij|.
    (c , s) = sym.schur2(A, p, q)
    A = J(p, q, θ)^T A J(p, q, θ)
    V = V J(p, q, θ)
end
```

Since $|a_{pq}|$ is the largest off-diagonal entry, off$(A)^2 \leq N(a_{pq}^2 + a_{qp}^2)$ where

$N = n(n-1)/2$. From (8.4.2) it follows that

$$\text{off}(B)^2 \le \left(1 - \frac{1}{N}\right) \text{off}(A)^2 .$$

By induction, if $A^{(k)}$ denotes the matrix A after k Jacobi updates, then

$$\text{off}(A^{(k)})^2 \le \left(1 - \frac{1}{N}\right)^k \text{off}(A^{(0)})^2.$$

This implies that the classical Jacobi procedure converges at a linear rate.

However, the asymptotic convergence rate of the method is considerably better than linear. Schonhage (1964) and van Kempen (1966) show that for k large enough, there is a constant c such that

$$\text{off}(A^{(k+N)}) \le c \cdot \text{off}(A^{(k)})^2$$

i.e., quadratic convergence. An earlier paper by Henrici (1958) established the same result for the special case when A has distinct eigenvalues. In the convergence theory for the Jacobi iteration, it is critical that $|\theta| \le \pi/4$. Among other things this precludes the possibility of "interchanging" nearly converged diagonal entries. This follows from the formulae $b_{pp} = a_{pp} - ta_{pq}$ and $b_{qq} = a_{qq} + ta_{pq}$, which can be derived from equations (8.4.1) and the definition $t = \sin(\theta)/\cos(\theta)$.

It is customary to refer to N Jacobi updates as a *sweep*. Thus, after a sufficient number of iterations, quadratic convergence is observed when examining off(A) after every sweep.

Example 8.4.1 Applying the classical Jacobi iteration to

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$$

we find

sweep	$O(\text{off}(A))$
0	10^{2}
1	10^{1}
2	10^{-2}
3	10^{-11}
4	10^{-17}

There is no rigorous theory that enables one to predict the number of sweeps that are required to achieve a specified reduction in off(A). However, Brent and Luk (1985) have argued heuristically that the number of sweeps is proportional to $\log(n)$ and this seems to be the case in practice.

8.4.4 The Cyclic-by-Row Algorithm

The trouble with the classical Jacobi method is that the updates involve $O(n)$ flops while the search for the optimal (p,q) is $O(n^2)$. One way to address this imbalance is to fix the sequence of subproblems to be solved in advance. A reasonable possibility is to step through all the subproblems in row-by-row fashion. For example, if $n = 4$ we cycle as follows:

$$(p,q) = (1,2),(1,3),(1,4),(2,3),(2,4),(3,4),(1,2),\ldots$$

This ordering scheme is referred to as *cyclic-by-row* and it results in the following procedure:

Algorithm 8.4.3 (Cyclic Jacobi) Given a symmetric $A \in \mathbb{R}^{n\times n}$ and a tolerance $tol > 0$, this algorithm overwrites A with V^TAV where V is orthogonal and $\text{off}(V^TAV) \le tol\| A \|_F$.

```
V = I_n
eps = tol|| A ||_F
while off(A) > eps
      for p = 1:n - 1
            for q = p + 1:n
                  (c , s) = sym.schur2(A, p, q)
                  A = J(p, q, θ)^T A J(p, q, θ)
                  V = V J(p, q, θ)
            end
      end
end
```

Cyclic Jacobi converges also quadratically. (See Wilkinson (1962) and van Kempen (1966).) However, since it does not require off-diagonal search, it is considerably faster than Jacobi's original algorithm.

Example 8.4.2 If the cyclic Jacobi method is applied to the matrix in Example 8.4.1 we find

Sweep	$O(\text{off}(A))$
0	10^2
1	10^1
2	10^{-1}
3	10^{-6}
4	10^{-16}

8.4.5 Error Analysis

Using Wilkinson's error analysis it is possible to show that if r sweeps are needed in Algorithm 8.4.3 then the computed d_i satisfy

$$\sum_{i=1}^{n}(d_i - \lambda_i)^2 \leq (\delta + k_r)\| A \|_F \mathbf{u}$$

for some ordering of A's eigenvalues λ_i. The parameter k_r depends mildly on r.

Although the cyclic Jacobi method converges quadratically, it is not generally competitive with the symmetric QR algorithm. For example, if we just count flops, then 2 sweeps of Jacobi is roughly equivalent to a complete QR reduction to diagonal form with accumulation of transformations. However, for small n this liability is not very dramatic. Moreover, if an approximate eigenvector matrix V is known, then V^TAV is almost diagonal, a situation that Jacobi can exploit but not QR.

Another interesting feature of the Jacobi method is that it can a compute the eigenvalues with small *relative* error if A is positive definite. To appreciate this point, note that the Wilkinson analysis cited above coupled the §8.1 perturbation theory ensures that the computed eigenvalues $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_n$ satisfy

$$\frac{|\hat{\lambda}_i - \lambda_i(A)|}{\lambda_i(A)} \approx \mathbf{u}\frac{\| A \|_2}{\lambda_i(A)} \leq \mathbf{u}\kappa_2(A).$$

However, a refined, componentwise error analysis by Demmel and Veselič (1992) shows that in the positive definite case,

$$\frac{|\hat{\lambda}_i - \lambda_i(A)|}{\lambda_i(A)} \approx \mathbf{u}\kappa_2(D^{-1}AD^{-1}). \qquad (8.4.4)$$

where $D = \text{diag}(\sqrt{a_{11}}, \ldots, \sqrt{a_{nn}})$ and this is generally a much smaller approximating bound. The key to establishing this result is some new perturbation theory and a demonstration that if A_+ is a computed Jacobi update obtained from the current matrix A_c, then the eigenvalues of A_+ are relatively close to the eigenvalues of A_c in the sense of (8.4.4). To make the whole thing work in practice, the termination criteria is not based upon the comparison of off(A) with $\mathbf{u}\| A \|_F$ but rather on the size of each $|a_{ij}|$ compared to $\mathbf{u}\sqrt{a_{ii}a_{jj}}$. This work is typical of a new genre of research concerned with high-accuracy algorithms based upon careful, componentwise error analysis. See Mathias (1995).

8.4.6 Parallel Jacobi

Perhaps the most interesting distinction between the QR and Jacobi approaches to the symmetric eigenvalue problem is the rich inherent paral-

lelism of the latter algorithm. To illustrate this, suppose $n = 4$ and group the six subproblems into three *rotation sets* as follows:

$$\begin{aligned} rot.set(1) &= \{(1,2),(3,4)\} \\ rot.set(2) &= \{(1,3),(2,4)\} \\ rot.set(3) &= \{(1,4),(2,3)\} \end{aligned}$$

Note that all the rotations within each of the three rotation sets are "nonconflicting." That is, subproblems (1,2) and (3,4) can be carried out in parallel. Likewise the (1,3) and (2,4) subproblems can be executed in parallel as can subproblems (1,4) and (2,3). In general, we say that

$$(i_1,j_1),(i_2,j_2),\ldots,(i_N,j_N) \qquad N = (n-1)n/2$$

is a *parallel ordering* of the set $\{(i,j) \mid 1 \le i < j \le n\}$ if for $s = 1{:}n-1$ the rotation set $rot.set(s) = \{\ (i_r,j_r) : r = 1 + n(s-1)/2{:}ns/2\ \}$ consists of nonconflicting rotations. This requires n to be even, which we assume throughout this section. (The odd n case can be handled by bordering A with a row and column of zeros and being careful when solving the subproblems that involve these augmented zeros.)

A good way to generate a parallel ordering is to visualize a chess tournament with n players in which everybody must play everybody else exactly once. In the $n = 8$ case this entails 7 "rounds." During round one we have the following four games:

1	3	5	7
2	4	6	8

$$rot.set(1) = \{\ (1,2),(3,4),(5,6),(7,8)\ \}$$

i.e., 1 plays 2, 3 plays 4, etc. To set up rounds 2 through 7, player 1 stays put and players 2 through 8 embark on a merry-go-round:

1	2	3	5
4	6	8	7

$$rot.set(2) = \{(1,4),(2,6),(3,8),(5,7)\}$$

1	4	2	3
6	8	7	5

$$rot.set(3) = \{(1,6),(4,8),(2,7),(3,5)\}$$

1	6	4	2
8	7	5	3

$$rot.set(4) = \{(1,8),(6,7),(4,5),(2,3)\}$$

1	8	6	4
7	5	3	2

$$rot.set(5) = \{(1,7),(5,8),(3,6),(2,4)\}$$

1	7	8	6
5	3	2	4

$$rot.set(6) = \{(1,5),(3,7),(2,8),(4,6)\}$$

1	5	7	8
3	2	4	6

$$rot.set(7) \;=\; \{(1,3),(2,5),(4,7),(6,8)\}$$

We can encode these operations in a pair of integer vectors $top(1{:}n/2)$ and $bot(1{:}n/2)$. During a given round $top(k)$ plays $bot(k)$, $k = 1{:}n/2$. The pairings for the next round is obtained by updating *top* and *bot* as follows:

```
function: [new.top, new.bot] = music(top, bot, n)
    m = n/2
    for k = 1:m
        if k = 1
            new.top(1) = 1
        else if k = 2
            new.top(k) = bot(1)
        elseif k > 2
            new.top(k) = top(k − 1)
        end
        if k = m
            new.bot(k) = top(k)
        else
            new.bot(k) = bot(k + 1)
        end
    end
```

Using **music** we obtain the following parallel order Jacobi procedure.

Algorithm 8.4.4 (Parallel Order Jacobi) Given a symmetric $A \in \mathbb{R}^{n\times n}$ and a tolerance $tol > 0$, this algorithm overwrites A with V^TAV where V is orthogonal and $\text{off}(V^TAV) \le tol\| A \|_F$. It is assumed that n is even.

```
V = I_n
eps = tol|| A ||_F
top = 1:2:n; bot = 2:2:n
while off(A) > eps
    for set = 1:n − 1
        for k = 1:n/2
            p = min(top(k), bot(k))
            q = max(top(k), bot(k))
            (c , s) = sym.schur2(A, p, q)
            A = J(p, q, θ)^T A J(p, q, θ)
            V = V J(p, q, θ)
        end
        [top, bot] = music(top, bot, n)
    end
end
```

Notice that the k-loop steps through $n/2$ independent, nonconflicting subproblems.

8.4.7 A Ring Procedure

We now discuss how Algorithm 8.4.4 could be implemented on a ring of p processors. We assume that $p = n/2$ for clarity. At any instant, Proc(μ) houses two columns of A and the corresponding V columns. For example, if $n = 8$ then here is how the column distribution of A proceeds from step to step:

	Proc(1)	Proc(2)	Proc(3)	Proc(4)
Step 1:	[1 2]	[3 4]	[5 6]	[7 8]
Step 2:	[1 4]	[2 6]	[3 8]	[5 7]
Step 3:	[1 6]	[4 8]	[2 7]	[3 5]
		etc.		

The ordered pairs denote the indices of the housed columns. The first index names the *left* column and the second index names the *right* column. Thus, the *left* and *right* columns in Proc(3) during step 3 are 2 and 7 respectively.

Note that in between steps, the columns are shuffled according to the permutation implicit in **music** and that nearest neighbor communication prevails. At each step, each processor oversees a single subproblem. This involves (a) computing an orthogonal $V_{small} \in \mathbb{R}^{2\times 2}$ that solves a local 2-by-2 Schur problem, (b) using the 2-by-2 V_{small} to update the two housed columns of A and V, (c) sending the 2-by-2 V_{small} to all the other processors, and (d) receiving the V_{small} matrices from the other processors and updating the local portions of A and V accordingly. Since A is stored by column, communication is necessary to carry out the V_{small} updates because they effect rows of A. For example, in the second step of the $n = 8$ problem, Proc(2) must receive the 2-by-2 rotations associated with subproblems (1,4), (3,8), and (5,7). These come from Proc(1), Proc(3), and Proc(4) respectively. In general, the sharing of the rotation matrices can be conveniently implemented by circulating the 2-by-2 V_{small} matrices in "merry go round" fashion around the ring. Each processor copies a passing 2-by-2 V_{small} into its local memory and then appropriately updates the locally housed portions of A and V.

The termination criteria in Algorithm 8.4.4 poses something of a problem in a distributed memory environment in that the value of off($\cdot$) and $\| A \|_F$ require access to all of A. However, these global quantities can be computed during the V matrix merry-go-round phase. Before the circulation of the V's begins, each processor can compute its contribution to $\| A \|_F$ and off($\cdot$) . These quantities can then be summed by each processor if they are placed on the merry-go-round and read at each stop. By the end of one revolution each processor has its own copy of $\| A \|_F$ and off($\cdot$).

8.4.8 Block Jacobi Procedures

It is usually the case when solving the symmetric eigenvalue problem on a p-processor machine that $n \gg p$. In this case a block version of the Jacobi algorithm may be appropriate. Block versions of the above procedures are straightforward. Suppose that $n = rN$ and that we partition the n-by-n matrix A as follows:

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & & \vdots \\ A_{N1} & \cdots & A_{NN} \end{bmatrix}.$$

Here, each A_{ij} is r-by-r. In block Jacobi the (p,q) subproblem involves computing the $2r$-by-$2r$ Schur decomposition

$$\begin{bmatrix} V_{pp} & V_{pq} \\ V_{qp} & V_{qq} \end{bmatrix}^T \begin{bmatrix} A_{pp} & A_{pq} \\ A_{qp} & A_{qq} \end{bmatrix} \begin{bmatrix} V_{pp} & V_{pq} \\ V_{qp} & V_{qq} \end{bmatrix} = \begin{bmatrix} D_{pp} & O \\ O & D_{qq} \end{bmatrix}$$

and then applying to A the block Jacobi rotation made up of the V_{ij} . If we call this block rotation V then it is easy to show that

$$\text{off}(V^TAV)^2 = \text{off}(A)^2 - \left(2\| A_{pq} \|_F^2 + \text{off}(A_{pp})^2 + \text{off}(A_{qq})^2\right).$$

Block Jacobi procedures have many interesting computational aspects. For example, there are many ways to solve the subproblems and the choice appears to be critical. See Bischof (1987).

Problems

P8.4.1 Let the scalar γ be given along with the matrix

$$A = \begin{bmatrix} w & x \\ x & z \end{bmatrix}.$$

It is desired to compute an orthogonal matrix

$$J = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

such that the (1, 1) entry of J^TAJ equals γ. Show that this requirement leads to the equation

$$(w-\gamma)\tau^2 - 2x\tau + (z-\gamma) = 0,$$

where $\tau = c/s$. Verify that this quadratic has real roots if γ satisfies $\lambda_2 \le \gamma \le \lambda_1$, where λ_1 and λ_2 are the eigenvalues of A.

P8.4.2 Let $A \in \mathbb{R}^{n\times n}$ be symmetric. Give an algorithm that computes the factorization

$$Q^TAQ = \gamma I + F$$

where Q is a product of Jacobi rotations, $\gamma = \text{trace}(A)/n$, and F has zero diagonal entries. Discuss the uniqueness of Q.

P8.4.3 Formulate Jacobi procedures for (a) skew symmetric matrices and (b) complex

Hermitian matrices.

P8.4.4 Partition the n-by-n real symmetric matrix A as follows:

$$A = \begin{bmatrix} a & v^T \\ v & A_1 \end{bmatrix} \begin{matrix} 1 \\ n-1 \end{matrix}$$
$$\quad 1 \quad n-1$$

Let Q be a Householder matrix such that if $B = Q^T A Q$, then $B(3{:}n, 1) = 0$. Let $J = J(1, 2, \theta)$ be determined such that if $C = J^T B J$, then $c_{12} = 0$ and $c_{11} \geq c_{22}$. Show $c_{11} \geq a + \| v \|_2$. La Budde (1964) formulated an algorithm for the symmetric eigenvalue probem based upon repetition of this Householder-Jacobi computation.

P8.4.5 Organize function **music** so that it involves minimum workspace.

P8.4.6 When implementing cyclic Jacobi, it is sensible to skip the annihilation of a_{pq} if its modulus is less than some small, sweep-dependent parameter, because the net reduction in off(A) is not worth the cost. This leads to what is called the *threshold Jacobi method*. Details concerning this variant of Jacobi's algorithm may be found in Wilkinson (1965, p.277). Show that appropriate thresholding can guarantee convergence.

Notes and References for Sec. 8.4

Jacobi's original paper is one of the earliest references found in the numerical analysis literature

C.G.J. Jacobi (1846). "Uber ein Leichtes Verfahren Die in der Theorie der Sacularstroungen Vorkommendern Gleichungen Numerisch Aufzulosen," *Crelle's J. 30,* 51–94.

Prior to the QR algorithm, the Jacobi technique was the standard method for solving dense symmetric eigenvalue problems. Early attempts to improve upon it include

M. Lotkin (1956). "Characteristic Values of Arbitrary Matrices," *Quart. Appl. Math. 14,* 267–75.
D.A. Pope and C. Tompkins (1957). "Maximizing Functions of Rotations: Experiments Concerning Speed of Diagonalization of Symmetric Matrices Using Jacobi's Method," *J. ACM 4,* 459–66.
C.D. La Budde (1964). "Two Classes of Algorithms for Finding the Eigenvalues and Eigenvectors of Real Symmetric Matrices," *J. ACM 11,* 53–58.

The computational aspects of Jacobi method are described in Wilkinson (1965,p.265). See also

H. Rutishauser (1966). "The Jacobi Method for Real Symmetric Matrices," *Numer. Math. 9,* 1–10. See also Wilkinson and Reinsch (1971, pp. 202–11).
N. Mackey (1995). "Hamilton and Jacobi Meet Again: Quaternions and the Eigenvalue Problem," *SIAM J. Matrix Anal. Applic. 16,* 421–435.

The method is also useful when a nearly diagonal matrix must be diagonalized. See

J.H. Wilkinson (1968). "Almost Diagonal Matrices with Multiple or Close Eigenvalues," *Lin. Alg. and Its Applic. 1,* 1–12.

Establishing the quadratic convergence of the classical and cyclic Jacobi iterations has attracted much attention:

P. Henrici (1958). "On the Speed of Convergence of Cyclic and Quasicyclic Jacobi Methods for Computing the Eigenvalues of Hermitian Matrices," *SIAM J. Appl. Math. 6,* 144–62.
E.R. Hansen (1962). "On Quasicyclic Jacobi Methods," *ACM J. 9,* 118–35.

J.H. Wilkinson (1962). "Note on the Quadratic Convergence of the Cyclic Jacobi Process," *Numer. Math. 6*, 296–300.
E.R. Hansen (1963). "On Cyclic Jacobi Methods," *SIAM J. Appl. Math. 11*, 448–59.
A. Schonhage (1964). "On the Quadratic Convergence of the Jacobi Process," *Numer. Math. 6*, 410–12.
H.P.M. van Kempen (1966). "On Quadratic Convergence of the Special Cyclic Jacobi Method," *Numer. Math. 9*, 19–22.
P. Henrici and K. Zimmermann (1968). "An Estimate for the Norms of Certain Cyclic Jacobi Operators," *Lin. Alg. and Its Applic. 1*, 489–501.
K.W. Brodlie and M.J.D. Powell (1975). "On the Convergence of Cyclic Jacobi Methods," *J. Inst. Math. Applic. 15*, 279–87.

Detailed error analyses that establish imprtant componentwise error bounds include

J. Barlow and J. Demmel (1990). "Computing Accurate Eigensystems of Scaled Diagonally Dominant Matrices," *SIAM J. Numer. Anal. 27*, 762–791.
J.W. Demmel and K. Veselić (1992). "Jacobi's Method is More Accurate than QR," *SIAM J. Matrix Anal. Appl. 13*, 1204–1245.
Z. Drmač (1994). *The Generalized Singular Value Problem*, Ph.D. Thesis, FernUniversitat, Hagen, Germany.
W.F. Mascarenhas (1994). "A Note on Jacobi Being More Accurate than QR," *SIAM J. Matrix Anal. Appl. 15*, 215–218.
R. Mathias (1995). "Accurate Eigensystem Computations by Jacobi Methods," *SIAM J. Matrix Anal. Appl. 16*, 977–1003.

Attempts have been made to extend the Jacobi iteration to other classes of matrices and to push through corresponding convergence results. The case of normal matrices is discussed in

H.H. Goldstine and L.P. Horowitz (1959). "A Procedure for the Diagonalization of Normal Matrices," *J. Assoc. Comp. Mach. 6*, 176–95.
G. Loizou (1972). "On the Quadratic Convergence of the Jacobi Method for Normal Matrices," *Comp. J. 15*, 274–76.
A. Ruhe (1972). "On the Quadratic Convergence of the Jacobi Method for Normal Matrices," *BIT 7*, 305–13.

See also

M.H.C. Paardekooper (1971). "An Eigenvalue Algorithm for Skew Symmetric Matrices," *Numer. Math. 17*, 189–202.
D. Hacon (1993). "Jacobi's Method for Skew-Symmetric Matrices," *SIAM J. Matrix Anal. Appl. 14*, 619–628.

Essentially, the analysis and algorithmic developments presented in the text carry over to the normal case with minor modification. For non-normal matrices, the situation is considerably more difficult. Consult

J. Greenstadt (1955). "A Method for Finding Roots of Arbitrary Matrices," *Math. Tables and Other Aids to Comp. 9*, 47–52.
C.E. Froberg (1965). "On Triangularization of Complex Matrices by Two Dimensional Unitary Tranformations," *BIT 5*, 230–34.
J. Boothroyd and P.J. Eberlein (1968). "Solution to the Eigenproblem by a Norm-Reducing Jacobi-Type Method (Handbook)," *Numer. Math. 11*, 1–12. See also Wilkinson and Reinsch (1971, pp.327–38).
A. Ruhe (1968). On the Quadratic Convergence of a Generalization of the Jacobi Method to Arbitrary Matrices," *BIT 8*, 210–31.
A. Ruhe (1969). "The Norm of a Matrix After a Similarity Transformation," *BIT 9*, 53–58.

P.J. Eberlein (1970). "Solution to the Complex Eigenproblem by a Norm-Reducing Jacobi-type Method," *Numer. Math.* *14*, 232-45. See also Wilkinson and Reinsch (1971, pp.404–17).

C.P. Huang (1975). "A Jacobi-Type Method for Triangularizing an Arbitrary Matrix," *SIAM J. Num. Anal. 12*, 566–70.

V. Hari (1982). "On the Global Convergence of the Eberlein Method for Real Matrices," *Numer. Math. 39*, 361–370.

G.W. Stewart (1985). "A Jacobi-Like Algorithm for Computing the Schur Decomposition of a Nonhermitian Matrix," *SIAM J. Sci. and Stat. Comp. 6*, 853–862.

W-W. Lin and C.W. Chen (1991). "An Acceleration Method for Computing the Generalized Eigenvalue Problem on a Parallel Computer," *Lin.Alg. and Its Applic. 146*, 49–65.

Jacobi methods for complex symmetric matrices have also been developed. See

J.J. Seaton (1969). "Diagonalization of Complex Symmetric Matrices Using a Modified Jacobi Method," *Comp. J. 12*, 156–57.

P.J. Eberlein (1971). "On the Diagonalization of Complex Symmetric Matrices," *J. Inst. Math. Applic. 7*, 377–83.

P. Anderson and G. Loizou (1973). "On the Quadratic Convergence of an Algorithm Which Diagonalizes a Complex Symmetric Matrix," *J. Inst. Math. Applic. 12*, 261–71.

P. Anderson and G. Loizou (1976). "A Jacobi-Type Method for Complex Symmetric Matrices (Handbook)," *Numer. Math. 25*, 347–63.

Although the symmetric QR algorithm is generally much faster than the Jacobi method, there are special settings where the latter technique is of interest. As we illustrated, on a parallel-computer it is possible to perform several rotations concurrently, thereby accelerating the reduction of the off-diagonal elements. See

A. Sameh (1971). "On Jacobi and Jacobi-like Algorithms for a Parallel Computer," *Math. Comp. 25*, 579–90.

J.J. Modi and J.D. Pryce (1985). "Efficient Implementation of Jacobi's Diagonalization Method on the DAP," *Numer. Math. 46*, 443-454.

D.S. Scott, M.T. Heath, and R.C. Ward (1986). "Parallel Block Jacobi Eigenvalue Algorithms Using Systolic Arrays," *Lin. Alg. and Its Applic. 77*, 345–356.

P.J. Eberlein (1987). "On Using the Jacobi Method on a Hypercube," in *Hypercube Multiprocessors*, ed. M.T. Heath, SIAM Publications, Philadelphia.

G. Shroff and R. Schreiber (1989). "On the Convergence of the Cyclic Jacobi Method for Parallel Block Orderings," *SIAM J. Matrix Anal. Appl. 10*, 326–346.

M.H.C. Paardekooper (1991). "A Quadratically Convergent Parallel Jacobi Process for Diagonally Dominant Matrices with Nondistinct Eigenvalues," *Lin.Alg. and Its Applic. 145*, 71 88.

8.5 Tridiagonal Methods

In this section we develop special methods for the symmetric tridiagonal eigenproblem. The tridiagonal form

$$T = \begin{bmatrix} a_1 & b_1 & & \cdots & 0 \\ b_1 & a_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & b_{n-1} \\ 0 & \cdots & & b_{n-1} & a_n \end{bmatrix} \tag{8.5.1}$$

can be obtained by Householder reduction (cf. §8.3.1). However, symmetric tridiagonal eigenproblems arise naturally in many settings.

We first discuss bisection methods that are of interest when selected portions of the eigensystem are required. This is followed by the presentation of a divide and conquer algorithm that can be used to acquire the full symmetric Schur decomposition in a way that is amenable to parallel processing.

8.5.1 Eigenvalues by Bisection

Let T_r denote the leading r-by-r principal submatrix of the matrix T in (8.5.1). Define the polynomials $p_r(x) = \det(T_r - xI)$, $r = 1{:}n$. A simple determinantal expansion shows that

$$p_r(x) = (a_r - x)p_{r-1}(x) - b_{r-1}^2 p_{r-2}(x) \tag{8.5.2}$$

for $r = 2{:}n$ if we set $p_0(x) = 1$. Because $p_n(x)$ can be evaluated in $O(n)$ flops, it is feasible to find its roots using the method of bisection. For example, if $p_n(y)p_n(z) < 0$ and $y < z$, then the iteration

```
while |y − z| > ε(|y| + |z|)
    x = (y + z)/2
    if p_n(x)p_n(y) < 0
        z = x
    else
        y = x
    end
end
```

is guaranteed to terminate with $(y + z)/2$ an approximate zero of $p_n(x)$, i.e., an approximate eigenvalue of T. The iteration converges linearly in that the error is approximately halved at each step.

8.5.2 Sturm Sequence Methods

Sometimes it is necessary to compute the kth largest eigenvalue of T for some prescribed value of k. This can be done efficiently by using the bisection idea and the following classical result:

Theorem 8.5.1 (Sturm Sequence Property) *If the tridiagonal matrix in (8.5.1) has no zero subdiagonal entries, then the eigenvalues of T_{r-1} strictly separate the eigenvalues of T_r:*

$$\lambda_r(T_r) < \lambda_{r-1}(T_{r-1}) < \lambda_{r-1}(T_r) < \cdots < \lambda_2(T_r) < \lambda_1(T_{r-1}) < \lambda_1(T_r).$$

Moreover, if $a(\lambda)$ denotes the number of sign changes in the sequence

$$\{\, p_0(\lambda),\, p_1(\lambda), \ldots,\, p_n(\lambda)\,\}$$

then $a(\lambda)$ equals the number of T's eigenvalues that are less than λ. Here, the polynomials $p_r(x)$ are defined by (8.5.2) and we have the convention that $p_r(\lambda)$ has the opposite sign of $p_{r-1}(\lambda)$ if $p_r(\lambda) = 0$.

Proof. It follows from Theorem 8.1.7 that the eigenvalues of T_{r-1} weakly separate those of T_r. To prove that the separation must be strict, suppose that $p_r(\mu) = p_{r-1}(\mu) = 0$ for some r and μ. It then follows from (8.5.2) and the assumption that T is unreduced that $p_0(\mu) = p_1(\mu) = \cdots = p_r(\mu) = 0$, a contradiction. Thus, we must have strict separation.

The assertion about $a(\lambda)$ is established in Wilkinson (1965, 300-301). We mention that if $p_r(\lambda) = 0$, then its sign is assumed to be opposite the sign of $p_{r-1}(\lambda)$. □

Example 8.5.1 If

$$T = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix}$$

then $\lambda(T) \approx \{.254,\ 1.82,\ 3.18,\ 4.74\}$. The sequence

$$\{\, p_0(2),\, p_1(2),\, p_2(2),\, p_3(2),\, p_4(2)\,\} \;=\; \{\, 1,\, -1,\, -1,\, 0,\, 1\,\}$$

confirms that there are two eigenvalues less than $\lambda = 2$.

Suppose we wish to compute $\lambda_k(T)$. From the Gershgorin theorem (Theorem 8.1.3) it follows that $\lambda_k(T) \in [y, z]$ where

$$y = \min_{1\le i\le n} a_i - |b_i| - |b_{i-1}| \qquad z = \max_{1\le i\le n} a_i + |b_i| + |b_{i-1}|$$

if we define $b_0 = b_n = 0$. With these starting values, it is clear from the Sturm sequence property that the iteration

```
while |z − y| > u(|y| + |z|)
      x = (y + z)/2
      if a(x) ≥ n − k                                        (8.5.3)
            z = x
      else
            y = x
      end
end
```

produces a sequence of subintervals that are repeatedly halved in length but which always contain $\lambda_k(T)$.

Example 8.5.2 If (8.5.3) is applied to the matrix of Example 8.5.1 with $k = 3$, then the values shown in the following table are generated:

y	z	x	$a(x)$
0.0000	5.0000	2.5000	2
0.0000	2.5000	1.2500	1
1.2500	2.5000	1.3750	1
1.3750	2.5000	1.9375	2
1.3750	1.9375	1.6563	1
1.6563	1.9375	1.7969	1

We conclude from the output that $\lambda_3(T) \in [\,1.7969,\ 1.9375\,]$. Note: $\lambda_3(T) \approx 1.82$.

During the execution of (8.5.3), information about the location of other eigenvalues is obtained. By systematically keeping track of this information it is possible to devise an efficient scheme for computing "contiguous" subsets of $\lambda(T)$, e.g., $\lambda_k(T), \lambda_{k+1}(T), \ldots, \lambda_{k+j}(T)$. See Barth, Martin, and Wilkinson (1967).

If selected eigenvalues of a general symmetric matrix A are desired, then it is necessary first to compute the tridiagonalization $T = U_0^T T U_0$ before the above bisection schemes can be applied. This can be done using Algorithm 8.3.1 or by the Lanczos algorithm discussed in the next chapter. In either case, the corresponding eigenvectors can be readily found via inverse iteration since tridiagonal systems can be solved in $O(n)$ flops. See §4.3.6 and §8.2.2.

In those applications where the original matrix A already has tridiagonal form, bisection computes eigenvalues with small relative error, regardless of their magnitude. This is in contrast to the tridiagonal QR iteration, where the computed eigenvalues $\tilde{\lambda}_i$ can be guaranteed only to have small absolute error: $|\tilde{\lambda}_i - \lambda_i(T)| \approx \mathbf{u}\| T \|_2$

Finally, it is possible to compute specific eigenvalues of a symmetric matrix by using the LDL^T factorization (see §4.2) and exploiting the Sylvester inertia theorem (Theorem 8.1.17). If

$$A - \mu I \;=\; LDL^T \qquad A = A^T \in \mathbb{R}^{n\times n}$$

is the LDL^T factorization of $A - \mu I$ with $D = \text{diag}(d_1, \ldots, d_n)$, then the number of negative d_i equals the number of $\lambda_i(A)$ that are less than μ. See Parlett (1980, p.46) for details.

8.5.3 Eigensystems of Diagonal Plus Rank-1 Matrices

Our next method for the symmetric tridiagonal eigenproblem requires that we be able to compute efficiently the eigenvalues and eigenvectors of a matrix of the form $D + \rho zz^T$ where $D \in \mathbb{R}^{n\times n}$is diagonal, $z \in \mathbb{R}^z$, and $\rho \in \mathbb{R}$. This problem is important in its own right and the key computations rest upon the following pair of results.

Lemma 8.5.2 *Suppose* $D = \text{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n\times n}$ *has the property that* $d_1 > \cdots > d_n$. *Assume that* $\rho \neq 0$ *and that* $z \in \mathbb{R}^n$ *has no zero components. If*

$$(D + \rho zz^T)v = \lambda v \qquad v \neq 0$$

then $z^Tv \neq 0$ *and* $D - \lambda I$ *is nonsingular.*

Proof. If $\lambda \in \lambda(D)$, then $\lambda = d_i$ for some i and thus

$$0 = e_i^T[(D - \lambda I)v + \rho(z^Tv)z] = \rho(z^Tv)z_i.$$

Since ρ and z_i are nonzero we must have $0 = z^Tv$ and so $Dv = \lambda v$. However, D has distinct eigenvalues and therefore, $v \in \text{span}\{e_i\}$. But then $0 = z^Tv = z_i$, a contradiction. Thus, D and $D + \rho zz^T$ do not have any common eigenvalues and $z^Tv \neq 0$. □

Theorem 8.5.3 *Suppose* $D = \text{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n\times n}$ *and that the diagonal entries satisfy* $d_1 > \cdots > d_n$. *Assume that* $\rho \neq 0$ *and that* $z \in \mathbb{R}^n$ *has no zero components. If* $V \in \mathbb{R}^{n\times n}$ *is orthogonal such that*

$$V^T(D + \rho zz^T)V = \text{diag}(\lambda_1, \ldots, \lambda_n)$$

with $\lambda_1 \geq \cdots \geq \lambda_n$ *and* $V = [\, v_1, \ldots, v_n \,]$, *then*

(a) *The* λ_i *are the* n *zeros of* $f(\lambda) = 1 + \rho z^T(D - \lambda I)^{-1}z$.

(b) *If* $\rho > 0$, *then* $\lambda_1 > d_1 > \lambda_2 > \cdots > d_n$.
If $\rho < 0$, *then* $d_1 > \lambda_1 > d_2 > \cdots > d_n > \lambda_n$.

(c) *The eigenvector* v_i *is a multiple of* $(D - \lambda_i I)^{-1}z$.

Proof. If $(D + \rho zz^T)v = \lambda v$, then

$$(D - \lambda I)v + \rho(z^Tv)z = 0. \tag{8.5.4}$$

We know from Lemma 8.5.2 that $D - \lambda I$ is nonsingular. Thus,

$$v \in \text{span}\{(D - \lambda I)^{-1} z\}$$

thereby establishing (c). Moreover, if we apply $z^T(D - \lambda I)^{-1}$ to both sides of equation (8.5.4) we obtain

$$z^T v \left(1 + \rho z^T (D - \lambda I)^{-1} z\right) \;=\; 0\,.$$

By Lemma 8.5.2, $z^T v \neq 0$ and so this shows that if $\lambda \in \lambda(D + \rho z z^T)$, then $f(\lambda) = 0$. We must show that all the zeros of f are eigenvalues of $D + \rho z z^T$ and that the interlacing relations (b) hold.

To do this we look more carefully at the equations

$$\begin{aligned} f(\lambda) &= 1 + \rho\left(\frac{z_1^2}{d_1 - \lambda} + \cdots + \frac{z_n^2}{d_n - \lambda}\right) \\ f'(\lambda) &= \rho\left(\frac{z_1^2}{(d_1 - \lambda)^2} + \cdots + \frac{z_n^2}{(d_n - \lambda)^2}\right) \end{aligned}$$

Note that f is monotone in between its poles. This allows us to conclude that if $\rho > 0$, then f has precisely n roots, one in each of the intervals

$$(d_n, d_{n-1}), \ldots, (d_2, d_1), (d_1, \infty).$$

If $\rho < 0$ then f has exactly n roots, one in each of the intervals

$$(-\infty, d_n), (d_n, d_{n-1}), \ldots, (d_2, d_1).$$

In either case, it follows that the zeros of f are precisely the eigenvalues of $D + \rho v v^T$.□

The theorem suggests that to compute V we (a) find the roots $\lambda_1, \ldots, \lambda_n$ of f using a Newton-like procedure and then (b) compute the columns of V by normalizing the vectors $(D - \lambda_i I)^{-1} z$ for $i = 1{:}n$. The same plan of attack can be followed even if there are repeated d_i and zero z_i.

Theorem 8.5.4 *If $D = \text{diag}(d_1, ..., d_n)$ and $z \in \mathbb{R}^n$, then there exists an orthogonal matrix V_1 such that if $V_1^T D V_1 \;=\; \text{diag}(\mu_1, \ldots, \mu_n)$ and $w = V_1^T z$ then*

$$\mu_1 > \mu_2 > \cdots > \mu_r \geq \mu_{r+1} \geq \cdots \geq \mu_n\,,$$

$w_i \neq 0$ for $i = 1{:}r$, and $w_i = 0$ for $i = r + 1{:}n$.

Proof. We give a constructive proof based upon two elementary operations. (a) Suppose $d_i = d_j$ for some $i < j$. Let $J(i, j, \theta)$ be a Jacobi rotation in the (i, j) plane with the property that the jth component of $J(i, j, \theta)^T z$ is zero. It is not hard to show that $J(i, j, \theta)^T D J(i, j, \theta) = D$. Thus, we can zero a component of z if there is a repeated d_i. (b) If $z_i = 0$,

$z_j \neq 0$, and $i < j$, then let P be the identity with columns i and j interchanged. It follows that P^TDP is diagonal, $(P^Tz)_i \neq 0$, and $(P^Tz)_j = 0$. Thus, we can permute all the zero z_i to the "bottom." Clearly, repetition of (a) and (b) eventually renders the desired canonical structure. V_1 is the product of the rotations. □

See Barlow (1993) and the references therein for a discussion of the solution procedures that we have outlined above.

8.5.4 A Divide and Conquer Method

We now present a divide-and-conquer method for computing the Schur decomposition

$$Q^TTQ = \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n) \qquad Q^TQ = I \tag{8.5.5}$$

for tridiagonal T that involves (a) "tearing" T in half, (b) computing the the Schur decompositions of the two parts, and (c) combining the two half-sized Schur decompositions into the required full size Schur decomposition. The overall procedure, developed by Dongarra and Sorensen (1987), is suitable for parallel computation.

We first show how T can be "torn" in half with a rank-one modification. For simplicity, assume $n = 2m$. Define $v \in \mathbb{R}^n$ as follows

$$v = \begin{bmatrix} e_m^{(m)} \\ \theta e_1^{(m)} \end{bmatrix}. \tag{8.5.6}$$

Note that for all $\rho \in \mathbb{R}$ the matrix $\tilde{T} = T - \rho vv^T$ is identical to T except in its "middle four" entries:

$$\tilde{T}(m{:}m+1, m{:}m+1) = \begin{bmatrix} a_m - \rho & b_m - \rho\theta \\ b_m - \rho\theta & a_{m+1} - \rho\theta^2 \end{bmatrix}.$$

If we set $\rho\theta = b_m$ then

$$T = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} + \rho vv^T$$

where

$$T_1 = \begin{bmatrix} a_1 & b_1 & & \cdots & 0 \\ b_1 & a_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & b_{m-1} \\ 0 & \cdots & & b_{m-1} & \tilde{a}_m \end{bmatrix},$$

$$T_2 = \begin{bmatrix} \tilde{a}_{m+1} & b_{m+1} & & \cdots & 0 \\ b_{m+1} & a_{m+2} & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & b_{n-1} \\ 0 & \cdots & & b_{n-1} & a_n \end{bmatrix},$$

and $\tilde{a}_m = a_m - \rho$ and $\tilde{a}_{m+1} = a_{m+1} - \rho\theta^2$.

Now suppose that we have m-by-m orthogonal matrices Q_1 and Q_2 such that $Q_1^T T_1 Q_1 = D_1$ and $Q_2^T T_2 Q_2 = D_2$ are each diagonal. If we set

$$U = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix},$$

then

$$U^T T U = U^T \left(\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} + \rho v v^T \right) U = D + \rho z z^T$$

where

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$$

is diagonal and

$$z = U^T v = \begin{bmatrix} Q_1^T e_m \\ \theta Q_2^T e_1 \end{bmatrix}.$$

Comparing these equations we see that the effective synthesis of the two half-sized Schur decompositions requires the quick and stable computation of an orthogonal V such that

$$V^T (D + \rho z z^T) V = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$$

which we discussed in §8.5.3.

8.5.5 A Parallel Implementation

Having stepped through the tearing and synthesis operations, we are ready to illustrate the overall process and how it can be implemented on a multiprocessor. For clarity, assume that $n = 8N$ for some positive integer N and that three levels of tearing are performed. We can depict this with a binary tree as shown in FIG. 8.5.1. The indices are specified in binary. FIG. 8.5.2 depicts a single node and should be interpreted to mean that the eigensystem for the tridiagonal $T(b)$ is obtained from the eigensystems of the tridiagonals $T(b0)$ and $T(b1)$. For example, the eigensystems for the N-by-N matrices $T(110)$ and $T(111)$ are combined to produce the eigensystem for the $2N$-by-$2N$ tridiagonal matrix $T(11)$.

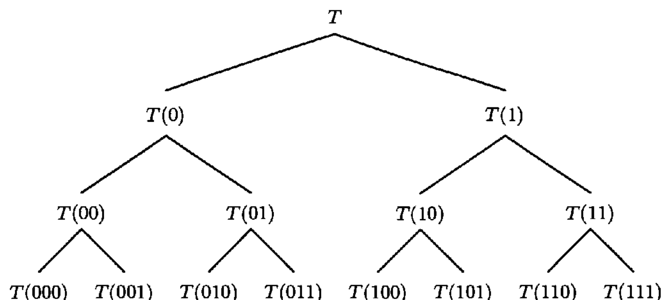


FIGURE 8.5.1 *Computation Tree*

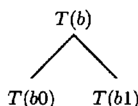


FIGURE 8.5.2 *Synthesis at a Node*

With tree-structured algorithms there is always the danger that parallelism is lost as the tree is "climbed" towards the root, but this is not the case in our problem. To see this suppose we have 8 processors and that the first task of Proc(b) is to compute the Schur decomposition of $T(b)$ where $b = 000, 001, 010, 011, 100, 101, 110, 111$. This portion of the computation is perfectly load balanced and does not involve interprocessor communication. (We are ignoring the Theorem 8.5.4 deflations, which are unlikely to cause significant load imbalance.)

At the next level there are four gluing operations to perform: $T(00)$, $T(01)$, $T(10)$, $T(11)$. However, each of these computations neatly subdivides and we can assign two processors to each task. For example, once the secular equation that underlies the $T(00)$ synthesis is known to both Proc(000) and Proc(001), then they each can go about getting half of the eigenvalues and corresponding eigenvectors. Likewise, 4 processors can each be assigned to the $T(0)$ and $T(1)$ problem. All 8 processors can participate in computing the eigensystem of T. Thus, at every level full parallelism

can be maintained because the eigenvalue/eigenvector computations are independent of one another.

Problems

P8.5.1 Suppose λ is an eigenvalue of a symmetric tridiagonal matrix T. Show that if λ has algebraic multiplicity k, then at least $k-1$ of T's subdiagonal elements are zero.

P8.5.2 Give an algorithm for determining ρ and θ in (8.5.6) with the property that $\theta \in \{-1, 1\}$ and $\min\{\, |a_r - \rho|, |a_{r+1} - \rho| \,\}$ is maximized.

P8.5.3 Let $p_r(\lambda) = \det(T(1{:}r, 1{:}r) - \lambda I_r)$ where T is given by (8.5.1). Derive a recursion for evaluating $p_n'(\lambda)$ and use it to develop a Newton iteration that can compute eigenvalues of T.

P8.5.4 What communication is necessary between the processors assigned to a particular T_b? Is it possible to share the work associated with the processing of repeated d_i and zero z_i ?

P8.5.5 If T is positive definite, does it follow that the matrices T_1 and T_2 in §8.5.4 are positive definite?

P8.5.6 Suppose that

$$A = \begin{bmatrix} D & v \\ v^T & d_{nn} \end{bmatrix}$$

where $D = \text{diag}(d_1, \ldots, d_{n-1})$ has distinct diagonal entries and $v \in \mathbb{R}^{n-1}$ has no zero entries. (a) Show that if $\lambda \in \lambda(A)$, then $D - \lambda I_{n-1}$ is nonsingular. (b) Show that if $\lambda \in \lambda(A)$, then λ is a zero of

$$f(\lambda) = \lambda + \sum_{k=1}^{n-1} \frac{v_k^2}{d_k - \lambda} - d_n.$$

P8.5.7 Suppose $A = S + \sigma u u^T$ where $S \in \mathbb{R}^{n\times n}$ is skew-symmetric, $u \in \mathbb{R}^n$, and $\sigma \in \mathbb{R}$. Show how to compute an orthogonal Q such that $Q^T A Q = T + \sigma e_1 e_1^T$ where T is tridiagonal and skew-symmetric and e_1 is the first column of I_n.

P8.5.8 It is known that $\lambda \in \lambda(T)$ where $T \in \mathbb{R}^{n\times n}$ is symmetric and tridiagonal with no zero subdiagonal entries. Show how to compute $x(1{:}n-1)$ from the equation $Tx = \lambda x$ given that $x_n = 1$.

Notes and References for Sec. 8.5

Bisection/ Strum sequence methods are discussed in

W. Barth, R.S. Martin, and J.H. Wilkinson (1967). "Calculation of the Eigenvalues of a Symmetric Tridiagonal Matrix by the Method of Bisection," *Numer. Math. 9*, 386–93. See also Wilkinson and Reinsch (1971, 249–256).

K.K. Gupta (1972). "Solution of Eigenvalue Problems by Sturm Sequence Method," *Int. J. Numer. Meth. Eng. 4*, 379–404.

Various aspects of the divide and conquer algorithm discussed in this section is detailed in

G.H. Golub (1973). "Some Modified Matrix Eigenvalue Problems," *SIAM Review 15*, 318–44.

J.R. Bunch, C.P. Nielsen, and D.C. Sorensen (1978). "Rank-One Modification of the Symmetric Eigenproblem," *Numer. Math. 31*, 31–48.

J.J.M. Cuppen (1981). "A Divide and Conquer Method for the Symmetric Eigenproblem," *Numer. Math. 36*, 177–95.

J.J. Dongarra and D.C. Sorensen (1987). "A Fully Parallel Algorithm for the Symmetric Eigenvalue Problem," *SIAM J. Sci. and Stat. Comp. 8*, S139–S154.
S. Crivelli and E.R. Jessup (1995). "The Cost of Eigenvalue Computation on Distributed Memory MIMD Computers," *Parallel Computing 21*, 401–422.

The very delicate computations required by the method are carefully analyzed in

J.L. Barlow (1993). "Error Analysis of Update Methods for the Symmetric Eigenvalue Problem," *SIAM J. Matrix Anal. Appl. 14*, 598–618.

Various generalizations to banded symmetric eigenproblems have been explored.

P. Arbenz, W. Gander, and G.H. Golub (1988). "Restricted Rank Modification of the Symmetric Eigenvalue Problem: Theoretical Considerations," *Lin. Alg. and Its Applic. 104*, 75–95.
P. Arbenz and G.H. Golub (1988). "On the Spectral Decomposition of Hermitian Matrices Subject to Indefinite Low Rank Perturbations with Applications," *SIAM J. Matrix Anal. Appl. 9*, 40–58.

A related divide and conquer method based on the "arrowhead" matrix (see P8.5.7) is given in

M. Gu and S.C. Eisenstat (1995). "A Divide-and-Conquer Algorithm for the Symmetric Tridiagonal Eigenproblem," *SIAM J. Matrix Anal. Appl. 16*, 172–191.

8.6 Computing the SVD

There are important relationships between the singular value decomposition of a matrix A and the Schur decompositions of the symmetric matrices A^TA, AA^T, and $\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$. Indeed, if

$$U^TAV = \text{diag}(\sigma_1, \ldots, \sigma_n)$$

is the SVD of $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$, then

$$V^T(A^TA)V = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2) \in \mathbb{R}^{n\times n} \tag{8.6.1}$$

and

$$U^T(AA^T)U = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2, \underbrace{0, \ldots, 0}_{m-n}) \in \mathbb{R}^{m\times m} \tag{8.6.2}$$

Moreover, if

$$U = [\, \underset{n}{U_1} \quad \underset{m-n}{U_2} \,]$$

and we define the orthogonal matrix $Q \in \mathbb{R}^{(m+n)\times(m+n)}$ by

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} V & V & 0 \\ U_1 & -U_1 & \sqrt{2}\,U_2 \end{bmatrix}$$

then

$$Q^T \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix} Q = \text{diag}(\sigma_1, \ldots, \sigma_n, -\sigma_1, \ldots, -\sigma_n, \underbrace{0, \ldots, 0}_{m-n}). \quad (8.6.3)$$

These connections to the symmetric eigenproblem allow us to adapt the mathematical and algorithmic developments of the previous sections to the singular value problem. Good references for this section include Lawson and Hanson (1974) and Stewart and Sun (1990).

8.6.1 Perturbation Theory and Properties

We first establish perturbation results for the SVD based on the theorems of §8.1. Recall that $\sigma_i(A)$ denotes the ith largest singular value of A.

Theorem 8.6.1 *If $A \in \mathbb{R}^{m\times n}$, then for $k = 1{:}\min\{m, n\}$*

$$\sigma_k(A) = \max_{\substack{\dim(S)=k \\ \dim(T)=k}} \min_{\substack{x\in S \\ y\in T}} \frac{y^TAx}{\| x \|_2 \| y \|_2} = \max_{\dim(S)=k} \min_{x\in S} \frac{\| Ax \|_2}{\| x \|_2}.$$

Note that in this expression $S \subseteq \mathbb{R}^n$ and $T \subseteq \mathbb{R}^m$ are subspaces.

Proof. The right-most characterization follows by applying Theorem 8.1.2 to A^TA. The remainder of the proof we leave as an exercise. □

Corollary 8.6.2 *If A and $A+E$ are in $\mathbb{R}^{m\times n}$ with $m \geq n$, then for $k = 1{:}n$*

$$|\sigma_k(A + E) - \sigma_k(A)| \leq \sigma_1(E) = \| E \|_2.$$

Proof. Apply Corollary 8.1.6 to

$$\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & (A+E)^T \\ A+E & 0 \end{bmatrix}. \; \square$$

Example 8.6.1 If

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \quad \text{and} \quad A+E = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6.01 \end{bmatrix}$$

then $\sigma(A) = \{9.5080, .7729\}$ and $\sigma(A+E) = \{9.5145, .7706\}$. It is clear that for $i = 1{:}2$ we have $|\sigma_i(A+E) - \sigma_i(A)| \leq \| E \|_2 = .01$.

Corollary 8.6.3 *Let $A = [\, a_1, \ldots, a_n \,] \in \mathbb{R}^{m\times n}$ be a column partitioning with $m \geq n$. If $A_r = [\, a_1, \ldots, a_r \,]$, then for $r = 1{:}n-1$*

$$\sigma_1(A_{r+1}) \geq \sigma_1(A_r) \geq \sigma_2(A_{r+1}) \geq \cdots \geq \sigma_r(A_{r+1}) \geq \sigma_r(A_r) \geq \sigma_{r+1}(A_{r+1}).$$

Proof. Apply Corollary 8.1.7 to A^TA. □

This last result says that by adding a column to a matrix, the largest singular value increases and the smallest singular value is diminished.

Example 8.3.2

$$A = \begin{bmatrix} 1 & 6 & 11 \\ 2 & 7 & 12 \\ 3 & 8 & 13 \\ 4 & 9 & 14 \\ 5 & 10 & 15 \end{bmatrix} \quad \Rightarrow \quad \begin{cases} \sigma(A_1) = \{7.4162\} \\ \sigma(A_2) = \{19.5377, 1.8095\} \\ \sigma(A_3) = \{35.1272, 2.4654, 0.0000\} \end{cases}$$

thereby confirming Corollary 8.6.3.

The next result is a Wielandt-Hoffman theorem for singular values:

Theorem 8.6.4 *If A and $A+E$ are in $\mathbb{R}^{m\times n}$ with $m \geq n$, then*

$$\sum_{k=1}^{n} (\sigma_k(A+E) - \sigma_k(A))^2 \;\leq\; \| E \|_F^2 .$$

Proof. Apply Theorem 8.1.4 to $\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & (A+E)^T \\ A+E & 0 \end{bmatrix}$. □

Example 8.6.3 If

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \quad \text{and} \quad A+E = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6.01 \end{bmatrix}$$

then

$$\sum_{k=1}^{2} (\sigma_k(A+E) - \sigma_k(A))^2 = .472 \times 10^{-4} \leq 10^{-4} = \| E \|_F^2 .$$

See Example 8.6.1.

For $A \in \mathbb{R}^{m\times n}$ we say that the k-dimensional subspaces $S \subseteq \mathbb{R}^n$ and $T \subseteq \mathbb{R}^m$ form a *singular subspace pair* if $x \in S$ and $y \in T$ imply $Ax \in T$ and $A^Ty \in S$. The following result is concerned with the perturbation of singular subspace pairs.

Theorem 8.6.5 *Let $A, E \in \mathbb{R}^{m\times n}$ with $m \geq n$ be given and suppose that $V \in \mathbb{R}^{n\times n}$ and $U \in \mathbb{R}^{m\times m}$ are orthogonal. Assume that*

$$V = \begin{matrix} [\; V_1 & V_2 \;] \\ r & n-r \end{matrix} \qquad U = \begin{matrix} [\; U_1 & U_2 \;] \\ r & m-r \end{matrix}$$

and that $\text{ran}(V_1)$ *and* $\text{ran}(U_1)$ *form a singular subspace pair for A. Let*

$$U^HAV = \begin{matrix} \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} & \begin{matrix} r \\ m-r \end{matrix} \\ \begin{matrix} r & n-r \end{matrix} & \end{matrix}$$

$$U^H E V = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{matrix} r \\ m-r \end{matrix}$$
$$\qquad\qquad r \quad n-r$$

and assume that

$$\delta = \min_{\substack{\sigma \in \sigma(A_{11}) \\ \gamma \in \sigma(A_{22})}} |\sigma - \gamma| > 0.$$

If

$$\| E \|_F \le \frac{\delta}{4},$$

then there exist matrices $P \in \mathbb{R}^{(n-r)\times r}$ *and* $Q \in \mathbb{R}^{(m-r)\times r}$ *satisfying*

$$\left\| \begin{bmatrix} Q \\ P \end{bmatrix} \right\|_F \le 4 \frac{\| E \|_F}{\delta}$$

such that $\text{ran}(V_1 + V_2 Q)$ *and* $\text{ran}(U_1 + U_2 P)$ *is a singular subspace pair for* $A + E$.

Proof. See Stewart (1973), Theorem 6.4. □

Roughly speaking, the theorem says that $O(\epsilon)$ changes in A can alter a singular subspace by an amount ϵ/δ, where δ measures the separation of the relevant singular values.

Example 8.6.4 The matrix $A = \text{diag}(2.000, 1.001, .999) \in \mathbb{R}^{4\times 3}$ has singular subspace pairs $(\text{span}\{v_i\}, \text{span}\{u_i\})$ for $i = 1, 2, 3$ where $v_i = e_i^{(3)}$ and $u_i = e_i^{(4)}$. Suppose

$$A + E = \begin{bmatrix} 2.000 & .010 & .010 \\ .010 & 1.001 & .010 \\ .010 & .010 & .999 \\ .010 & .010 & .010 \end{bmatrix}$$

The corresponding columns of the matrices

$$\hat{U} = [\,\hat{u}_1\, \hat{u}_2\, \hat{u}_3] = \begin{bmatrix} .9999 & -.0144 & .0007 \\ .0101 & .7415 & .6708 \\ .0101 & .6707 & -.7616 \\ .0051 & .0138 & -.0007 \end{bmatrix}$$

$$\hat{V} = [\,\hat{v}_1\, \hat{v}_2\, \hat{v}_3] = \begin{bmatrix} .9999 & -.0143 & .0007 \\ .0101 & .7416 & .6708 \\ .0101 & .6707 & -.7416 \end{bmatrix}$$

define singular subspace pairs for $A+E$. Note that the pair $\{\text{span}\{\hat{v}_i\}, \text{span}\{\hat{u}_i\}\}$, is close to $\{\text{span}\{v_i\}, \text{span}\{u_i\}\}$ for $i = 1$ but not for $i = 2$ or 3. On the other hand, the singular subspace pair $\{\text{span}\{\hat{v}_2, \hat{v}_3\}, \text{span}\{\hat{u}_2, \hat{u}_3\}\}$ is close to $\{\text{span}\ \{v_2, v_3\}, \text{span}\{u_2, u_3\}\}$.

8.6.2 The SVD Algorithm

We now show how a variant of the QR algorithm can be used to compute the SVD of an $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. At first glance, this appears straightforward. Equation (8.6.1) suggests that we

- form $C = A^TA$,
- use the symmetric QR algorithm to compute $V_1^TCV_1 = \text{diag}(\sigma_i^2)$,
- apply QR with column pivoting to AV_1 obtaining $U^T(AV_1)\Pi = R$.

Since R has orthogonal columns, it follows that $U^TA(V_1\Pi)$ is diagonal. However, as we saw in Example 5.3.2, the formation of A^TA can lead to a loss of information. The situation is not quite so bad here, since the original A is used to compute U.

A preferable method for computing the SVD is described in Golub and Kahan (1965). Their technique finds U and V simultaneously by *implicitly* applying the symmetric QR algorithm to A^TA. The first step is to reduce A to upper bidiagonal form using Algorithm 5.4.2:

$$U_B^TAV_B = \begin{bmatrix} B \\ 0 \end{bmatrix} \qquad B = \begin{bmatrix} d_1 & f_1 & & \cdots & 0 \\ 0 & d_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & f_{n-1} \\ 0 & \cdots & & 0 & d_n \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

The remaining problem is thus to compute the SVD of B. To this end, consider applying an implicit-shift QR step (Algorithm 8.3.2) to the tridiagonal matrix $T = B^TB$:

- Compute the eigenvalue λ of

$$T(m{:}n, m{:}n) = \begin{bmatrix} d_m^2 + f_{m-1}^2 & d_mf_m \\ d_mf_m & d_n^2 + f_m^2 \end{bmatrix} \qquad m = n-1$$

  that is closer to $d_n^2 + f_m^2$.

- Compute $c_1 = \cos(\theta_1)$ and $s_1 = \sin(\theta_1)$ such that

$$\begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix}^T \begin{bmatrix} d_1^2 - \lambda \\ d_1f_1 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$$

  and set $G_1 = G(1, 2, \theta_1)$.

- Compute Givens rotations $G_2, \ldots, G_{n-1}$ so that if $Q = G_1 \cdots G_{n-1}$ then $Q^T T Q$ is tridiagonal and $Qe_1 = G_1 e_1$.

Note that these calculations require the explicit formation of $B^T B$, which, as we have seen, is unwise from the numerical standpoint.

Suppose instead that we apply the Givens rotation G_1 above to B directly. Illustrating with the $n = 6$ case this gives

$$B \leftarrow BG_1 = \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ + & \times & \times & 0 & 0 & 0 \\ 0 & 0 & \times & \times & 0 & 0 \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}.$$

We then can determine Givens rotations U_1, V_2, $U_2, \ldots$, V_{n-1}, and U_{n-1} to chase the unwanted nonzero element down the bidiagonal:

$$B \leftarrow U_1^T B = \begin{bmatrix} \times & \times & + & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 \\ 0 & 0 & \times & \times & 0 & 0 \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$B \leftarrow BV_2 = \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 \\ 0 & + & \times & \times & 0 & 0 \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$B \leftarrow U_2^T B = \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ 0 & \times & \times & + & 0 & 0 \\ 0 & 0 & \times & \times & 0 & 0 \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

and so on. The process terminates with a new bidiagonal $\bar{B}$ that is related to B as follows:

$$\bar{B} = (U_{n-1}^T \cdots U_1^T) B (G_1 V_2 \cdots V_{n-1}) = \bar{U}^T B \bar{V}.$$

Since each V_i has the form $V_i = G(i, i+1, \theta_i)$ where $i = 2{:}n-1$, it follows that $\bar{V}e_1 = Qe_1$. By the implicit Q theorem we can assert that $\bar{V}$ and Q are essentially the same. Thus, we can implicitly effect the transition from T to $\bar{T} = \bar{B}^T\bar{B}$ by working directly on the bidiagonal matrix B.

Of course, for these claims to hold it is necessary that the underlying tridiagonal matrices be unreduced. Since the subdiagonal entries of B^TB are of the form d_{i-1}, f_i, it is clear that we must search the bidiagonal band for zeros. If $f_k = 0$ for some k, then

$$B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{matrix} k \\ n-k \end{matrix}$$
$$\qquad\quad k \quad n-k$$

and the original SVD problem decouples into two smaller problems involving the matrices B_1 and B_2. If $d_k = 0$ for some $k < n$, then premultiplication by a sequence of Givens transformations can zero f_k. For example, if $n = 6$ and $k = 3$, then by rotating in row planes (3,4), (3,5), and (3,6) we can zero the entire third row:

$$B = \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & 0 & 0 \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{(3,4)} \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & + & 0 \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$\xrightarrow{(3,5)} \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & + \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{(3,6)} \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

If $d_n = 0$, then the last column can be zeroed with a series of column rotations in planes $(n-1, n)$, $(n-2, n), \ldots, (1, n)$. Thus, we can decouple if $f_1 \cdots f_{n-1} = 0$ or $d_1 \cdots d_n = 0$.

Algorithm 8.6.1 (Golub-Kahan SVD Step) Given a bidiagonal matrix $B \in \mathbb{R}^{m\times n}$ having no zeros on its diagonal or superdiagonal, the following algorithm overwrites B with the bidiagonal matrix $\bar{B} = \bar{U}^T B\bar{V}$ where $\bar{U}$ and $\bar{V}$ are orthogonal and $\bar{V}$ is essentially the orthogonal matrix that would be obtained by applying Algorithm 8.3.2 to $T = B^TB$.

Let μ be the eigenvalue of the trailing 2-by-2 submatrix of $T = B^TB$ that is closer to t_{nn}.
$y = t_{11} - \mu$
$z = t_{12}$
for $k = 1{:}n-1$
  Determine $c = \cos(\theta)$ and $s = \sin(\theta)$ such that

$$\begin{bmatrix} y & z \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} * & 0 \end{bmatrix}$$

  $B = BG(k, k+1, \theta)$
  $y = b_{kk}$; $z = b_{k+1,k}$
  Determine $c = \cos(\theta)$ and $s = \sin(\theta)$ such that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix}$$

  $B = G(k, k+1, \theta)^T B$
  if $k < n-1$
    $y = b_{k,k+1}$; $z = b_{k,k+2}$
  end
end

An efficient implementation of this algorithm would store B's diagonal and superdiagonal in vectors $a(1{:}n)$ and $f(1{:}n-1)$ respectively and would require $30n$ flops and $2n$ square roots. Accumulating U requires $6mn$ flops. Accumulating V requires $6n^2$ flops.

Typically, after a few of the above SVD iterations, the superdiagonal entry f_{n-1} becomes negligible. Criteria for smallness within B's band are usually of the form

$$\begin{aligned} |f_i| &\leq \epsilon(\,|d_i| + |d_{i+1}|\,) \\ |d_i| &\leq \epsilon \,\|\, B \,\| \end{aligned}$$

where ϵ is a small multiple of the unit roundoff and $\|\cdot\|$ is some computationally convenient norm.

Combining Algorithm 5.4.2 (bidiagonalization), Algorithm 8.6.1, and the decoupling calculations mentioned earlier gives

Algorithm 8.6.2 (The SVD Algorithm) Given $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ and ϵ, a small multiple of the unit roundoff, the following algorithm overwrites A with $U^TAV = D + E$, where $U \in \mathbb{R}^{m\times n}$ is orthogonal, $V \in \mathbb{R}^{n\times n}$ is orthogonal, $D \in \mathbb{R}^{m\times n}$ is diagonal, and E satisfies $\|E\|_2 \approx \mathbf{u}\|A\|_2$.

Use Algorithm 5.4.2 to compute the bidiagonalization

$$\begin{bmatrix} B \\ 0 \end{bmatrix} \leftarrow (U_1 \cdots U_n)^T A (V_1 \cdots V_{n-2})$$

until $q = n$
  Set $b_{i,i+1}$ to zero if $|b_{i,i+1}| \le \epsilon(|b_{ii}| + |b_{i+1,i+1}|)$
    for any $i = 1{:}n-1$.
  Find the largest q and the smallest p such that if

$$B = \begin{bmatrix} B_{11} & 0 & 0 \\ 0 & B_{22} & 0 \\ 0 & 0 & B_{33} \end{bmatrix} \begin{matrix} p \\ n-p-q \\ q \end{matrix}$$
$$\qquad\quad p \qquad n-p-q \qquad q$$

    then B_{33} is diagonal and B_{22} has nonzero superdiagonal.
  if $q < n$
    if any diagonal entry in B_{22} is zero, then zero
      the superdiagonal entry in the same row.
    else
      Apply Algorithm 8.6.1 to B_{22},
      $B = \text{diag}(I_p, U, I_{q+m-n})^T B \text{diag}(I_p, V, I_q)$
    end
  end
end

The amount of work required by this algorithm and its numerical properties are discussed in §5.4.5 and §5.5.8.

Example 8.6.5 If Algorithm 8.6.2 is applied to

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

then the superdiagonal elements converge to zero as follows:

Iteration	$O(\|a_{21}\|)$	$O(\|a_{32}\|)$	$O(\|a_{43}\|)$
1	10^0	10^0	10^0
2	10^0	10^0	10^0
3	10^0	10^0	10^0
4	10^0	10^{-1}	10^{-2}
5	10^0	10^{-1}	10^{-8}
6	10^0	10^{-1}	10^{-27}
7	10^0	10^{-1}	converg.
8	10^0	10^{-4}	
9	10^{-1}	10^{-14}	
10	10^{-1}	converg.	
11	10^{-4}		
12	10^{-12}		
13	converg.		

Observe the cubic-like convergence.

8.6.3 Jacobi SVD Procedures

It is straightforward to adapt the Jacobi procedures of §8.4 to the SVD problem. Instead of solving a sequence of 2-by-2 symmetric eigenproblems, we solve a sequence of 2-by-2 SVD problems. Thus, for a given index pair (p, q) we compute a pair of rotations such that

$$\begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix}^T \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix} = \begin{bmatrix} d_p & 0 \\ 0 & d_q \end{bmatrix}.$$

See P8.6.8. The resulting algorithm is referred to as *two-sided* because each update involves a pre- and post-multiplication.

A *one-sided* Jacobi algorithm involves a sequence of pairwise column orthogonalizations. For a given index pair (p, q) a Jacobi rotation $J(p, q, \theta)$ is determined so that columns p and q of $AJ(p.q, \theta)$ are orthogonal to each other. See P8.6.8. Note that this corresponds to zeroing the (p, q) and (q, p) entries in A^TA. Once AV has sufficiently orthogonal columns, the rest of the SVD (U and Σ) follows from column scaling: $AV = U\Sigma$.

Problems

P8.6.1 Show that if $B \in \mathbb{R}^{n\times n}$ is an upper bidiagonal matrix having a repeated singular value, then B must have a zero on its diagonal or superdiagonal.

P8.6.2 Give formulae for the eigenvectors of $\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$ in terms of the singular vectors of $A \in \mathbb{R}^{m\times n}$ where $m \geq n$.

P8.6.3 Give an algorithm for reducing a complex matrix A to *real* bidiagonal form using complex Householder transformations.

P8.6.4 Relate the singular values and vectors of $A = B + iC$ ($B, C \in \mathbb{R}^{m\times n}$) to those of $\begin{bmatrix} B & -C \\ C & B \end{bmatrix}$.

P8.6.5 Complete the proof of Theorem 8.6.1.

P8.6.6 Assume that $n = 2m$ and that $S \in \mathbb{R}^{n\times n}$ is skew-symmetric and tridiagonal. Show that there exists a permutation $P \in \mathbb{R}^{n\times n}$ such that P^TSP has the following form:

$$P^TSP = \begin{bmatrix} 0 & -B^T \\ B & 0 \end{bmatrix} \begin{matrix} m \\ m \end{matrix} \quad .$$
$$\qquad\qquad\quad m \quad\; m$$

Describe B. Show how to compute the eigenvalues and eigenvectors of S via the SVD of B. Repeat for the case $n = 2m + 1$.

P8.6.7 (a) Let

$$C = \begin{bmatrix} w & x \\ y & z \end{bmatrix}$$

be real. Give a stable algorithm for computing c and s with $c^2 + s^2 = 1$ such that

$$B = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} C$$

is symmetric. (b) Combine (a) with the Jacobi trigonometric calculations in the text to obtain a stable algorithm for computing the SVD of C. (c) Part (b) can be used to

develop a Jacobi-like algorithm for computing the SVD of $A \in \mathbb{R}^{n\times n}$. For a given (p,q) with $p < q$, Jacobi transformations $J(p,q,\theta_1)$ and $J(p,q,\theta_2)$ are determined such that if

$$B = J(p,q,\theta_1)^T A J(p,q,\theta_2),$$

then $b_{pq} = b_{qp} = 0$. Show

$$\text{off}(B)^2 = \text{off}(A)^2 - b_{pq}^2 - b_{qp}^2.$$

How might p and q be determined? How could the algorithm be adapted to handle the case when $A \in \mathbb{R}^{m\times n}$ with $m > n$?

P8.6.8 Let x and y be in $\mathbb{R}^m$ and define the orthogonal matrix Q by

$$Q = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}.$$

Give a stable algorithm for computing c and s such that the columns of $[x,y]Q$ are orthogonal to each other.

P8.6.9 Suppose $B \in \mathbb{R}^{n\times n}$ is upper bidiagonal with $b_{nn} = 0$. Show how to construct orthogonal U and V (product of Givens rotations) so that U^TBV is upper bidiagonal with a zero nth column.

P8.6.10 Suppose $B \in \mathbb{R}^{n\times n}$ is upper bidiagonal with diagonal entries $d(1{:}n)$ and superdiagonal entries $f(1{:}n-1)$. State and prove a singular value version of Theorem 8.5.1.

Notes and References for Sec. 8.6

The mathematical properties of the SVD are discussed in Stewart and Sun (1990) as well as

A.R. Amir-Moez (1965). *Extremal Properties of Linear Transformations and Geometry of Unitary Spaces*, Texas Tech University Mathematics Series, no. 243, Lubbock, Texas.

G.W. Stewart (1973). "Error and Perturbation Bounds for Subspaces Associated with Certain Eigenvalue Problems," *SIAM Review 15*, 727–64.

P.A. Wedin (1972). "Perturbation Bounds in Connection with the Singular Value Decomposition," *BIT 12*, 99–111.

G.W. Stewart (1979). "A Note on the Perturbation of Singular Values," *Lin. Alg. and Its Applic. 28*, 213–16.

G.W. Stewart (1984). "A Second Order Perturbation Expansion for Small Singular Values," *Lin. Alg. and Its Applic. 56*, 231–236.

R.J. Vaccaro (1994). "A Second-Order Perturbation Expansion for the SVD," *SIAM J. Matrix Anal. Applic. 15*, 661–671.

The idea of adapting the symmetric QR algorithm to compute the SVD first appeared in

G.H. Golub and W. Kahan (1965). "Calculating the Singular Values and Pseudo-Inverse of a Matrix," *SIAM J. Num. Anal. Ser. B 2*, 205–24.

and then came some early implementations:

P.A. Businger and G.H. Golub (1969). "Algorithm 358: Singular Value Decomposition of a Complex Matrix," *Comm. Assoc. Comp. Mach. 12*, 564–65.

G.H. Golub and C. Reinsch (1970). "Singular Value Decomposition and Least Squares Solutions," *Numer. Math. 14*, 403–20. See also Wilkinson and Reinsch (1971, 134–51).

Interesting algorithmic developments associated with the SVD appear in

J.J.M. Cuppen (1983). "The Singular Value Decomposition in Product Form," *SIAM J. Sci. and Stat. Comp. 4,* 216–222.
J.J. Dongarra (1983). "Improving the Accuracy of Computed Singular Values," *SIAM J. Sci. and Stat. Comp. 4,* 712–719.
S. Van Huffel, J. Vandewalle, and A. Haegemans (1987). "An Efficient and Reliable Algorithm for Computing the Singular Subspace of a Matrix Associated with its Smallest Singular Values," *J. Comp. and Appl. Math. 19,* 313–330.
P. Deift, J. Demmel, L.-C. Li, and C. Tomei (1991). "The Bidiagonal Singular Value Decomposition and Hamiltonian Mechanics," *SIAM J. Num. Anal. 28,* 1463–1516.
R. Mathias and G.W. Stewart (1993). "A Block QR Algorithm and the Singular Value Decomposition," *Lin. Alg. and Its Applic. 182,* 91–100.
Å. Björck, E. Grimme, and P. Van Dooren (1994). "An Implicit Shift Bidiagonalization Algorithm for Ill-Posed Problems," *BIT 34,* 510–534.

The Polar decomposition of a matrix can be computed immediately from its SVD. However, special algorithms have been developed just for this purpose.

N.J. Higham (1986). "Computing the Polar Decomposition—with Applications," *SIAM J. Sci. and Stat. Comp. 7,* 1160–1174.
N.J. Higham and P. Papadimitriou (1994). "A Parallel Algorithm for Computing the Polar Decomposition," *Parallel Comp. 20,* 1161-1173.

Jacobi methods for the SVD fall into two categories. The two-sided Jacobi algorithms repeatedly perform the update $A \leftarrow U^T AV$ producing a sequence of iterates that are increasingly diagonal.

E.G. Kogbetliantz (1955). "Solution of Linear Equations by Diagonalization of Coefficient Matrix," *Quart. Appl. Math. 13,* 123–132.
G.E. Forsythe and P. Henrici (1960). "The Cyclic Jacobi Method for Computing the Principal Values of a Complex Matrix," *Trans. Amer. Math. Soc. 94,* 1–23.
C.C. Paige and P. Van Dooren (1986). "On the Quadratic Convergence of Kogbetliantz's Algorithm for Computing the Singular Value Decomposition," *Lin. Alg. and Its Applic. 77,* 301-313.
J.P. Charlier and P. Van Dooren (1987). "On Kogbetliantz's SVD Algorithm in the Presence of Clusters," *Lin. Alg. and Its Applic. 95,* 135–160.
Z. Bai (1988). "Note on the Quadratic Convergence of Kogbetliantz's Algorithm for Computing the Singular Value Decomposition," *Lin. Alg. and Its Applic. 104,* 131 140.
J.P. Charlier, M. Vanbegin, P. Van Dooren (1988). "On Efficient Implementation of Kogbetliantz's Algorithm for Computing the Singular Value Decomposition," *Numer. Math. 52,* 279–300.
K.V. Fernando (1989). "Linear Convergence of the Row Cyclic Jacobi and Kogbetliantz methods," *Numer. Math. 56,* 73–92.

The one-sided Jacobi SVD procedures repeatedly perform the update $A \leftarrow AV$ producing a sequence of iterates with columns that are increasingly orthogonal.

J.C. Nash (1975). "A One-Sided Tranformation Method for the Singular Value Decomposition and Algebraic Eigenproblem," *Comp. J. 18,* 74 76.
P.C. Hansen (1988). "Reducing the Number of Sweeps in Hestenes Method," in *Singular Value Decomposition and Signal Processing,* ed. E.F. Deprettere, North Holland.
K. Veselič and V. Hari (1989). "A Note on a One-Sided Jacobi Algorithm," *Numer. Math. 56,* 627–633.

Numerous parallel implementations have been developed.

F.T. Luk (1980). "Computing the Singular Value Decomposition on the ILLIAC IV," *ACM Trans. Math. Soft. 6,* 524–39.

R.P. Brent and F.T. Luk (1985). "The Solution of Singular Value and Symmetric Eigenvalue Problems on Multiprocessor Arrays," *SIAM J. Sci. and Stat. Comp. 6*, 69–84.
R.P. Brent, F.T. Luk, and C. Van Loan (1985). "Computation of the Singular Value Decomposition Using Mesh Connected Processors," *J. VLSI Computer Systems 1*, 242–270.
F.T. Luk (1986). "A Triangular Processor Array for Computing Singular Values," *Lin. Alg. and Its Applic. 77*, 259–274.
M. Berry and A. Sameh (1986). "Multiprocessor Jacobi Algorithms for Dense Symmetric Eigenvalue and Singular Value Decompositions," in *Proc. International Conference on Parallel Processing*, 433–440.
R. Schreiber (1986). "Solving Eigenvalue and Singular Value Problems on an Undersized Systolic Array," *SIAM J. Sci. and Stat. Comp. 7*, 441–451.
C.H. Bischof and C. Van Loan (1986). "Computing the SVD on a Ring of Array Processors," in *Large Scale Eigenvalue Problems*, eds. J. Cullum and R. Willoughby, North Holland, 51-66.
C.H. Bischof (1987). "The Two-Sided Block Jacobi Method on Hypercube Architectures," in *Hypercube Multiprocessors*, ed. M.T. Heath, SIAM Press, Philadelphia.
C.H. Bischof (1989). "Computing the Singular Value Decomposition on a Distributed System of Vector Processors," *Parallel Computing 11*, 171–186.
S. Van Huffel and H. Park (1994). "Parallel Tri- and Bidiagonalization of Bordered Bidiagonal Matrices," *Parallel Computing 20*, 1107–1128.
B. Lang (1996). "Parallel Reduction of Banded Matrices to Bidiagonal Form," *Parallel Computing 22*, 1–18.

The divide and conquer algorithms devised for for the symmetric eigenproblem have SVD analogs:

E.R. Jessup and D.C. Sorensen (1994). "A Parallel Algorithm for Computing the Singular Value Decomposition of a Matrix," *SIAM J. Matrix Anal. Appl. 15*, 530–548.
M. Gu and S.C. Eisenstat (1995). "A Divide-and-Conquer Algorithm for the Bidiagonal SVD," *SIAM J. Matrix Anal. Appl. 16*, 79–92.

Careful analyses of the SVD calculation include

J.W. Demmel and W. Kahan (1990). "Accurate Singular Values of Bidiagonal Matrices," *SIAM J. Sci. and Stat. Comp. 11*, 873–912.
K.V. Fernando and B.N. Parlett (1994). "Accurate Singular Values and Differential qd Algorithms," *Numer. Math. 67*, 191–230.
S. Chandrasekaren and I.C.F. Ipsen (1994). "Backward Errors for Eigenvalue and Singular Value Decompositions," *Numer. Math. 68*, 215–223.

High accuracy SVD calculation and connections among the Cholesky, Schur, and singular value computations are discussed in

J.W. Demmel and K. Veselič (1992). "Jacobi's Method is More Accurate than QR," *SIAM J. Matrix Anal. Appl. 13*, 1204–1245.
R. Mathias (1995). "Accurate Eigensystem Computations by Jacobi Methods," *SIAM J. Matrix Anal. Appl. 16*, 977–1003.

8.7 Some Generalized Eigenvalue Problems

Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a symmetric positive definite $B \in \mathbb{R}^{n\times n}$, we consider the problem of finding a nonzero vector x and a scalar λ so $Ax = \lambda Bx$. This is the *symmetric-definite generalized eigenproblem.* The scalar λ can be thought of as a *generalized eigenvalue.* As λ varies, $A - \lambda B$ defines a *pencil* and our job is to determine

$$\lambda(A,B) = \{\, \lambda \mid \det(A - \lambda B) = 0 \,\}.$$

A symmetric-definite generalized eigenproblem can be transformed to an equivalent problem with a congruence transformation:

$$A - \lambda B \text{ is singular} \quad \Leftrightarrow \quad (X^TAX) - \lambda(X^TBX) \text{ is singular}$$

Thus, if X is nonsingular, then $\lambda(A,B) = \lambda(X^TAX, X^TBX)$.

In this section we present various structure-preserving procedures that solve such eigenproblems through the careful selection of X. The related generalized singular value decomposition problem is also discussed.

8.7.1 Mathematical Background

We seek is a stable, efficient algorithm that computes X such that X^TAX and X^TBX are both in "canonical form." The obvious form to aim for is diagonal form.

Theorem 8.7.1 *Suppose A and B are n-by-n symmetric matrices, and define $C(\mu)$ by*

$$C(\mu) \;=\; \mu A + (1-\mu)B \qquad \mu \in \mathbb{R}\,. \tag{8.7.1}$$

If there exists a $\mu \in [0,1]$ such that $C(\mu)$ is non-negative definite and

$$\text{null}(C(\mu)) \;=\; \text{null}(A) \cap \text{null}(B)$$

then there exists a nonsingular X such that both X^TAX and X^TBX are diagonal.

Proof. Let $\mu \in [0,1]$ be chosen so that $C(\mu)$ is non-negative definite with the property that $\text{null}(C(\mu)) - \text{null}(A) \cap \text{null}(B)$. Let

$$Q_1^TC(\mu)Q_1 \;=\; \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \qquad D = \text{diag}(d_1,\ldots,d_k),\; d_i > 0$$

be the Schur decomposition of $C(\mu)$ and define $X_1 = Q_1\text{diag}(D^{-1/2}, I_{n-k})$. If $A_1 = X_1^TAX_1$, $B_1 = X_1^TBX_1$, and $C_1 - X_1^TC(\mu)X_1$, then

$$C_1 \;=\; \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} \;=\; \mu A_1 + (1-\mu)B_1.$$

Since $\text{span}\{e_{k+1}, \ldots, e_n\} = \text{null}(C_1) = \text{null}(A_1) \cap \text{null}(B_1)$ it follows that A_1 and B_1 have the following block structure:

$$A_1 = \begin{bmatrix} A_{11} & 0 \\ 0 & 0 \end{bmatrix} \begin{matrix} k \\ n-k \end{matrix} \qquad B_1 = \begin{bmatrix} B_{11} & 0 \\ 0 & 0 \end{bmatrix} \begin{matrix} k \\ n-k \end{matrix} .$$
$$\begin{matrix} k & n-k \end{matrix} \qquad\qquad \begin{matrix} k & n-k \end{matrix}$$

Moreover $I_k = \mu A_{11} + (1-\mu)B_{11}$.

Suppose $\mu \neq 0$. It then follows that if $Z^T B_{11} Z = \text{diag}(b_1, \ldots, b_k)$ is the Schur decomposition of B_{11} and we set $X = X_1 \text{diag}(Z, I_{n-k})$ then

$$X^T B X = \text{diag}(b_1, \ldots, b_k, 0, \ldots, 0) \equiv D_B$$

and

$$\begin{aligned} X^T A X &= \frac{1}{\mu} X^T \left(C(\mu) - (1-\mu)B\right) X \\ &= \frac{1}{\mu}\left(\begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} - (1-\mu)D_B\right) \equiv D_A . \end{aligned}$$

On the other hand, if $\mu = 0$, then let $Z^T A_{11} Z = \text{diag}(a_1, \ldots, a_k)$ be the Schur decomposition of A_{11} and set $X = X_1 \text{diag}(Z, I_{n-k})$. It is easy to verify that in this case as well, both $X^T A X$ and $X^T B X$ are diagonal. □

Frequently, the conditions in Theorem 8.7.1 are satisfied because either A or B is positive definite.

Corollary 8.7.2 *If $A - \lambda B \in \mathbb{R}^{n \times n}$ is symmetric-definite, then there exists a nonsingular $X = [\, x_1, \ldots, x_n \,]$ such that*

$$X^T A X = \text{diag}(a_1, \ldots, a_n) \quad \textit{and} \quad X^T B X = \text{diag}(b_1, \ldots, b_n) .$$

Moreover, $Ax_i = \lambda_i B x_i$ for $i = 1{:}n$ where $\lambda_i = a_i / b_i$.

Proof. By setting $\mu = 0$ in Theorem 8.7.1 we see that symmetric-definite pencils can be simultaneously diagonalized. The rest of the corollary is easily verified. □

Example 8.7.1 If

$$A = \begin{bmatrix} 229 & 163 \\ 163 & 116 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 81 & 59 \\ 59 & 43 \end{bmatrix}$$

then $A - \lambda B$ is symmetric-definite and $\lambda(A, B) = \{5, -1/2\}$. If

$$X = \begin{bmatrix} 3 & -5 \\ -4 & 7 \end{bmatrix}$$

then $X^TAX = \text{diag}(5,-1)$ and $X^TBX = \text{diag}(1,2)$.

Stewart (1979) has worked out a perturbation theory for symmetric pencils $A - \lambda B$ that satisfy

$$c(A,B) = \min_{\|x\|_2=1} (x^TAx)^2 + (x^TBx)^2 > 0 \qquad (8.7.2)$$

The scalar $c(A,B)$ is called the *Crawford number* of the pencil $A - \lambda B$.

Theorem 8.7.3 *Suppose $A - \lambda B$ is an n-by-n symmetric-definite pencil with eigenvalues*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n.$$

Suppose E_A and E_B are symmetric n-by-n matrices that satisfy

$$\epsilon^2 = \| E_A \|_2^2 + \| E_B \|_2^2 < c(A,B).$$

Then $(A + E_A) - \lambda(B + E_B)$ is symmetric-definite with eigenvalues

$$\mu_1 \geq \cdots \geq \mu_n$$

that satisfy

$$|\arctan(\lambda_i) - \arctan(\mu_i)| \leq \arctan(\epsilon/c(A,B))$$

for $i = 1{:}n$.

Proof. See Stewart (1979). □

8.7.2 Methods for the Symmetric-Definite Problem

Turning to algorithmic matters, we first present a method for solving the symmetric-definite problem that utilizes both the Cholesky factorization and the symmetric QR algorithm.

Algorithm 8.7.1 Given $A = A^T \in \mathbb{R}^{n\times n}$ and $B = B^T \in \mathbb{R}^{n\times n}$ with B positive definite, the following algorithm computes a nonsingular X such that $X^TBX = I_n$ and $X^TAX = \text{diag}(a_1, \ldots, a_n)$.

Compute the Cholesky factorization $B = GG^T$
    using Algorithm 4.2.2.
Compute $C = G^{-1}AG^{-T}$.
Use the symmetric QR algorithm to compute the Schur
    decomposition $Q^TCQ = \text{diag}(a_1, \ldots, a_n)$.
Set $X = G^{-T}Q$.

This algorithm requires about $14n^3$ flops. In a practical implementation, A can be overwritten by the matrix C. See Martin and Wilkinson (1968c) for details. Note that

$$\lambda(A,B) = \lambda(A,GG^T) = \lambda(G^{-1}AG^{-T},I) = \lambda(C) = \{a_1,\ldots,a_n\}.$$

If $\hat{a}_i$ is a computed eigenvalue obtained by Algorithm 8.7.1, then it can be shown that $\hat{a}_i \in \lambda(G^{-1}AG^{-T} + E_i)$, where $\| E_i \|_2 \approx \mathbf{u}\| A \|_2\| B^{-1} \|_2$. Thus, if B is ill-conditioned, then $\hat{a}_i$ may be severely contaminated with roundoff error even if a_i is a well-conditioned generalized eigenvalue. The problem, of course, is that in this case, the matrix $C = G^{-1}AG^{-T}$ can have some very large entries if B, and hence G, is ill-conditioned. This difficulty can sometimes be overcome by replacing the matrix G in Algorithm 8.7.1 with $VD^{-1/2}$ where $V^TBV = D$ is the Schur decomposition of B. If the diagonal entries of D are ordered from smallest to largest, then the large entries in C are concentrated in the upper left-hand corner. The small eigenvalues of C can then be computed without excessive roundoff error contamination (or so the heuristic goes). For further discussion, consult Wilkinson (1965, pp.337–38).

Example 8.7.2 If

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} .001 & 0 & 0 \\ 1 & .001 & 0 \\ 2 & 1 & .001 \end{bmatrix}$$

and $B = GG^T$, then the two smallest eigenvalues of $A - \lambda B$ are

$$a_1 = -0.619402940600584 \qquad a_2 = 1.627440079051887.$$

If 17-digit floating point arithmetic is used, then these eigenvalues are computed to full machine precision when the symmetric QR algorithm is applied to $fl(D^{-1/2}V^TAVD^{-1/2})$, where $B = VDV^T$ is the Schur decomposition of B. On the other hand, if Algorithm 8.7.1 is applied, then

$$\hat{a}_1 = -0.619373517376444 \qquad \hat{a}_2 = 1.627516601905228.$$

The reason for obtaining only four correct significant digits is that $\kappa_2(B) \approx 10^{18}$.

The condition of the matrix X in Algorithm 8.7.1 can sometimes be improved by replacing B with a suitable convex combination of A and B. The connection between the eigenvalues of the modified pencil and those of the original are detailed in the proof of Theorem 8.7.1.

Other difficulties concerning Algorithm 8.7.1 revolve around the fact that $G^{-1}AG^{-T}$ is generally full even when A and B are sparse. This is a serious problem, since many of the symmetric-definite problems arising in practice are large and sparse.

Crawford (1973) has shown how to implement Algorithm 8.7.1 effectively when A and B are banded. Aside from this case, however, the simultaneous diagonalization approach is impractical for the large, sparse symmetric-definite problem.

An alternative idea is to extend the Rayleigh quotient iteration (8.4.4) as follows:

x_0 given with $\| x_0 \|_2 = 1$
for $k = 0, 1, \ldots$
  $\mu_k = x_k^T A x_k / x_k^T B x_k$ (8.7.3)
  Solve $(A - \mu_k B) z_{k+1} = B x_k$ for z_{k+1}.
  $x_{k+1} = z_{k+1} / \| z_{k+1} \|_2$
end

The mathematical basis for this iteration is that

$$\lambda = \frac{x^T A x}{x^T B x} \tag{8.7.4}$$

minimizes

$$f(\lambda) = \| Ax - \lambda Bx \|_B \tag{8.7.5}$$

where $\|\cdot\|_B$ is defined by $\|z\|_B^2 = z^T B^{-1} z$. The mathematical properties of (8.7.3) are similar to those of (8.4.4). Its applicability depends on whether or not systems of the form $(A - \mu B)z = x$ can be readily solved. A similar comment pertains to the following generalized orthogonal iteration:

$Q_0 \in \mathbb{R}^{n \times p}$ given with $Q_0^T Q_0 = I_p$
for $k = 1, 2, \ldots$
  Solve $BZ_k = AQ_{k-1}$ for Z_k. (8.7.6)
  $Z_k = Q_k R_k$ (QR factorization)
end

This is mathematically equivalent to (7.3.4) with A replaced by $B^{-1}A$. Its practicality depends on how easy it is to solve linear systems of the form $Bz = y$.

Sometimes A and B are so large that neither (8.7.3) nor (8.7.6) can be invoked. In this situation, one can resort to any of a number of gradient and coordinate relaxation algorithms. See Stewart (1976) for an extensive guide to the literature.

8.7.3 The Generalized Singular Value Problem

We conclude with some remarks about symmetric pencils that have the form $A^T A - \lambda B^T B$ where $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times n}$. This pencil underlies the *generalized singular value decomposition* (GSVD), a decomposition that is useful in several constrained least squares problems. (Cf. §12.1.) Note that by Theorem 8.7.1 there exists a nonsingular $X \in \mathbb{R}^{n \times n}$ such that $X^T (A^T A) X$ and $X^T (B^T B) X$ are both diagonal. The value of the GSVD

is that these diagonalizations can be achieved without forming A^TA and B^TB.

Theorem 8.7.4 (Generalized Singular Value Decomposition) *If we have $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ and $B \in \mathbb{R}^{p\times n}$, then there exist orthogonal $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{p\times p}$ and an invertible $X \in \mathbb{R}^{n\times n}$ such that*

$$U^TAX = C = \text{diag}(c_1,\ldots,c_n) \qquad c_i \geq 0$$

and

$$V^TBX = S = \text{diag}(s_1,\ldots,s_q) \qquad s_i \geq 0$$

where $q = \min(p,n)$.

Proof. The proof of this decomposition appears in Van Loan (1976). We present a more constructive proof along the lines of Paige and Saunders (1981). For clarity we assume that $\text{null}(A) \cap \text{null}(B) = \{0\}$ and $p \geq n$. We leave it to the reader to extend the proof so that it covers theses cases.

Let

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R \qquad (8.7.6)$$

be a QR factorization with $Q_1 \in \mathbb{R}^{m\times n}$, $Q_2 \in \mathbb{R}^{p\times n}$, and $R \in \mathbb{R}^{n\times n}$. Paige and Saunders show that the SVD's of Q_1 and Q_2 are related in the sense that

$$Q_1 = UCW^T \qquad Q_2 = VSW^T \qquad (8.7.7)$$

Here, U,V, and W are orthogonal, $C = \text{diag}(c_i)$ with $0 \leq c_1 \leq \cdots \leq c_n$, $S = \text{diag}(s_i)$ with $s_1 \geq \cdots \geq s_n$, and $C^TC + S^TS = I_n$. The decomposition (8.7.7) is a variant of the CS decomposition in §2.6 and from it we conclude that $A = Q_1R = UC(W^TR)$ and $B = Q_2R = VS(W^TR)$. The theorem follows by setting $X = (W^TR)^{-1}$, $D_A = C$, and $D_B = S$. The invertibility of R follows from our assumption that $\text{null}(A) \cap \text{null}(B) = \{0\}$. □

The elements of the set $\sigma(A,B) \equiv \{ c_1/s_1,\ldots,c_n/s_q \}$ are referred to as the *generalized singular values* of A and B. Note that $\sigma \in \sigma(A,B)$ implies that $\sigma^2 \in \lambda(A^TA, B^TB)$. The theorem is a generalization of the SVD in that if $B = I_n$, then $\sigma(A,B) = \sigma(A)$.

Our proof of the GSVD is of practical importance since Stewart (1983) and Van Loan (1985) have shown how to stably compute the CS decomposition. The only tricky part is the inversion of W^TR to get X. Note that the columns of $X = [\, x_1,\ldots,x_n \,]$ satisfy

$$s_i^2A^TAx_i = c_i^2B^TBx_i \qquad i = 1{:}n$$

and so if $s_i \neq 0$ then $A^TAx_i = \sigma_i^2B^TBx_i$ where $\sigma_i = c_i/s_i$. Thus, the x_i are aptly termed the *generalized singular vectors* of the pair (A,B).

In several applications an orthonormal basis for some designated generalized singular vector subspace space span$\{x_{i_1},\ldots,x_{i_k}\}$ is required. We show how this can be accomplished without any matrix inversions or cross products:

- Compute the QR factorization

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R.$$

- Compute the CS decomposition

$$Q_1 = UCW^T \qquad Q_2 = VSW^T$$

  and order the diagonals of C and S so that

$$\{c_1/s_1,\ldots,c_k/s_k\} = \{c_{i_1}/s_{i_1},\ldots,c_{i_k}/s_{i_k}\}.$$

- Compute orthogonal Z and upper triangular T so $TZ = W^TR$. (See P8.7.5.) Note that if $X^{-1} = W^TR = TZ$, then $X = Z^TT^{-1}$ and so the first k rows of Z are an orthonormal basis for span$\{x_1,\ldots,x_k\}$.

Problems

P8.7.1 Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and $G \in \mathbb{R}^{n\times n}$ is lower triangular and nonsingular. Give an efficient algorithm for computing $C = G^{-1}AG^{-T}$.

P8.7.2 Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and $B \in \mathbb{R}^{n\times n}$ is symmetric positive definite. Give an algorithm for computing the eigenvalues of AB that uses the Cholesky factorization and the symmetric QR algorithm.

P8.7.3 Show that if C is real and diagonalizable, then there exist symmetric matrices A and B, B nonsingular, such that $C = AB^{-1}$. This shows that symmetric pencils $A-\lambda B$ are essentially general.

P8.7.4 Show how to convert an $Ax = \lambda Bx$ problem into a generalized singular value problem if A and B are both symmetric and non-negative definite.

P8.7.5 Given $Y \in \mathbb{R}^{n\times n}$ show how to compute Householder matrices $H_2,\ldots,H_n$ so that $YH_n\cdots H_2 = T$ is upper triangular. Hint: H_k zeros out the kth row.

P8.7.6 Suppose

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix} = \lambda\begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix}$$

where $A \in \mathbb{R}^{m\times n}$, $B_1 \in \mathbb{R}^{m\times m}$, and $B_2 \in \mathbb{R}^{n\times n}$. Assume that B_1 and B_2 are positive definite with Cholesky triangles G_1 and G_2 respectively. Relate the generalized eigenvalues of this problem to the singular values of $G_1^{-1}AG_2^{-T}$

P8.7.7 Suppose A and B are both symmetric positive definite. Show how to compute $\lambda(A,B)$ and the corresponding eigenvectors using the Cholesky factorization and CS decomposition.

Notes and References for Sec. 8.7

An excellent survey of computational methods for symmetric-definite pencils is given in

G.W. Stewart (1976). "A Bibliographical Tour of the Large Sparse Generalized Eigenvalue Problem," in *Sparse Matrix Computations* , ed., J.R. Bunch and D.J. Rose, Academic Press, New York.

Some papers of particular interest include

R.S. Martin and J.H. Wilkinson (1968c). "Reduction of a Symmetric Eigenproblem $Ax = \lambda Bx$ and Related Problems to Standard Form," *Numer. Math. 11,* 99–110.
G. Peters and J.H. Wilkinson (1969). "Eigenvalues of $Ax = \lambda Bx$ with Band Symmetric A and B," *Comp. J. 12,* 398-404.
G. Fix and R. Heiberger (1972). "An Algorithm for the Ill-Conditioned Generalized Eigenvalue Problem," *SIAM J. Num. Anal. 9,* 78 88.
C.R. Crawford (1973). "Reduction of a Band Symmetric Generalized Eigenvalue Problem," *Comm. ACM 16,* 41 44.
A. Ruhe (1974). "SOR Methods for the Eigenvalue Problem with Large Sparse Matrices," *Math. Comp.* 28, 695–710.
C.R. Crawford (1976). "A Stable Generalized Eigenvalue Problem," *SIAM J. Num. Anal. 13,* 854–60.
A. Bunse-Gerstner (1984). "An Algorithm for the Symmetric Generalized Eigenvalue Problem," *Lin. Alg. and Its Applic. 58,* 43–68.
C.R. Crawford (1986). "Algorithm 646 PDFIND: A Routine to Find a Positive Definite Linear Combination of Two Real Symmetric Matrices," *ACM Trans. Math. Soft. 12,* 278–282.
C.R. Crawford and Y.S. Moon (1983). "Finding a Positive Definite Linear Combination of Two Hermitian Matrices," *Lin. Alg. and Its Applic. 51,* 37–48.
W. Shougen and Z. Shuqin (1991). "An Algorithm for $Ax = \lambda Bx$ with Symmetric and Positive Definite A and B," *SIAM J. Matrix Anal. Appl. 12,* 654–660.
K. Li and T-Y. Li (1993). "A Homotopy Algorithm for a Symmetric Generalized Eigenproblem," *Numerical Algorithms 4,* 167–195.
K. Li, T-Y. Li, and Z. Zeng (1994). "An Algorithm for the Generalized Symmetric Tridiagonal Eigenvalue Problem," *Numerical Algorithms 8,* 269–291.
H. Zhang and W.F. Moss (1994). "Using Parallel Banded Linear System Solvers in Generalized Eigenvalue Problems," *Parallel Computing 20,* 1089–1106

The simultaneous reduction of two symmetric matrices to diagonal form is discussed in

A. Berman and A. Ben-Israel (1971). "A Note on Pencils of Hermitian or Symmetric Matrices," *SIAM J. Applic. Math. 21,* 51–54.
F. Uhlig (1973). "Simultaneous Block Diagonalization of Two Real Symmetric Matrices," *Lin. Alg. and Its Applic. 7,* 281–89.
F. Uhlig (1976). "A Canonical Form for a Pair of Real Symmetric Matrices That Generate a Nonsingular Pencil," *Lin. Alg. and Its Applic. 14,* 189–210.
K.N. Majinder (1979). "Linear Combinations of Hermitian and Real Symmetric Matrices," *Lin. Alg. and Its Applic. 25,* 95–105.

The perturbation theory that we presented for the symmetric-definite problem was taken from

G.W. Stewart (1979). "Perturbation Bounds for the Definite Generalized Eigenvalue Problem," *Lin. Alg. and Its Applic. 23,* 69–86.

See also

L. Elsner and J. Guang Sun (1982). "Perturbation Theorems for the Generalized Eigenvalue Problem,; *Lin. Alg. and its Applic. 48,* 341-357.
J. Guang Sun (1982). "A Note on Stewart's Theorem for Definite Matrix Pairs," *Lin. Alg. and Its Applic. 48,* 331–339.

J. Guang Sun (1983). "Perturbation Analysis for the Generalized Singular Value Problem," *SIAM J. Numer. Anal. 20*, 611–625.

C.C. Paige (1984). "A Note on a Result of Sun J.-Guang: Sensitivity of the CS and GSV Decompositions," *SIAM J. Numer. Anal. 21*, 186–191.

The generalized SVD and some of its applications are discussed in

C.F. Van Loan (1976). "Generalizing the Singular Value Decomposition," *SIAM J. Num. Anal. 13*, 76–83.

C.C. Paige and M. Saunders (1981). "Towards A Generalized Singular Value Decomposition," *SIAM J. Num. Anal. 18*, 398–405.

B. Kågström (1985). "The Generalized Singular Value Decomposition and the General $A - \lambda B$ Problem," *BIT 24*, 568-583.

Stable methods for computing the CS and generalized singular value decompositions are described in

G.W. Stewart (1983). "A Method for Computing the Generalized Singular Value Decomposition," in *Matrix Pencils* , ed. B. Kågström and A. Ruhe, Springer-Verlag, New York, pp. 207-20.

C.F. Van Loan (1985). "Computing the CS and Generalized Singular Value Decomposition," *Numer. Math. 46*, 479–492.

M.T. Heath, A.J. Laub, C.C. Paige, and R.C. Ward (1986). "Computing the SVD of a Product of Two Matrices," *SIAM J. Sci. and Stat. Comp. 7*, 1147-1159.

C.C. Paige (1986). "Computing the Generalized Singular Value Decomposition," *SIAM J. Sci. and Stat. Comp. 7*, 1126–1146.

L.M. Ewerbring and F.T. Luk (1989). "Canonical Correlations and Generalized SVD: Applications and New Algorithms," *J. Comput. Appl. Math. 27*, 37–52.

J. Erxiong (1990). "An Algorithm for Finding Generalized Eigenpairs of a Symmetric Definite Matrix Pencil," *Lin.Alg. and Its Applic. 132*, 65–91.

P.C. Hansen (1990). "Relations Between SVD and GSVD of Discrete Regularization Problems in Standard and General Form," *Lin.Alg. and Its Applic. 141*, 165–176.

H. Zha (1991). "The Restricted Singular Value Decomposition of Matrix Triplets," *SIAM J. Matrix Anal. Appl. 12*, 172–194.

B. De Moor and G.H. Golub (1991). "The Restricted Singular Value Decomposition: Properties and Applications," *SIAM J. Matrix Anal. Appl. 12*, 401–425.

V. Hari (1991). "On Pairs of Almost Diagonal Matrices," *Lin. Alg. and Its Applic. 148*, 193–223.

B. De Moor and P. Van Dooren (1992). "Generalizing the Singular Value and QR Decompositions," *SIAM J. Matrix Anal. Appl. 13*, 993–1014.

H. Zha (1992). "A Numerical Algorithm for Computing the Restricted Singular Value Decomposition of Matrix Triplets," *Lin.Alg. and Its Applic. 168*, 1–25.

R-C. Li (1993). "Bounds on Perturbations of Generalized Singular Values and of Associated Subspaces," *SIAM J. Matrix Anal. Appl. 14*, 195–234.

K. Veselič (1993). "A Jacobi Eigenreduction Algorithm for Definite Matrix Pairs," *Numer. Math. 64*, 241–268.

Z. Bai and H. Zha (1993). "A New Preprocessing Algorithm for the Computation of the Generalized Singular Value Decomposition," *SIAM J. Sci. Comp. 14*, 1007–1012.

L. Kaufman (1993). "An Algorithm for the Banded Symmetric Generalized Matrix Eigenvalue Problem," *SIAM J. Matrix Anal. Appl. 14*, 372–389.

G.E. Adams, A.W. Bojanczyk, and F.T. Luk (1994). "Computing the PSVD of Two 2×2 Triangular Matrices," *SIAM J. Matrix Anal. Appl. 15*, 366–382.

Z. Drmač (1994). *The Generalized Singular Value Problem*, Ph.D. Thesis, FernUniversitat, Hagen, Germany.

R-C. Li (1994). "On Eigenvalue Variations of Rayleigh Quotient Matrix Pencils of a Definite Pencil," *Lin. Alg. and Its Applic. 208/209*, 471–483.

Chapter 9

Lanczos Methods

In this chapter we develop the Lanczos method, a technique that can be used to solve certain large, sparse, symmetric eigenproblems $Ax = \lambda x$. The method involves partial tridiagonalizations of the given matrix A. However, unlike the Householder approach, no intermediate, full submatrices are generated. Equally important, information about A's extremal eigenvalues tends to emerge long before the tridiagonalization is complete. This makes the Lanczos algorithm particularly useful in situations where a few of A's largest or smallest eigenvalues are desired.

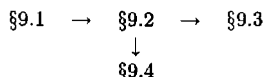
The derivation and exact arithmetic attributes of the method are presented in §9.1. The key aspects of the Kaniel-Paige theory are detailed. This theory explains the extraordinary convergence properties of the Lanczos process. Unfortunately, roundoff errors make the Lanczos method somewhat difficult to use in practice. The central problem is a loss of orthogonality among the Lanczos vectors that the iteration produces. There are several ways to cope with this as we discuss §9.2.

In §9.3 we show how the "Lanczos idea" can be applied to solve an assortment of singular value, least squares, and linear equations problems. Of particular interest is the development of the conjugate gradient method for symmetric positive definite linear systems. The Lanczos-conjugate gradient connection is explored further in the next chapter. In §9.4 we discuss the Arnoldi iteration which is based on the Hessenberg decomposition and a

version of the Lanczos process that can (sometimes) be used to tridiagonalize unsymmetric matrices.

Before You Begin

Chapters 5 and 8 are required for §9.1-9.3 and Chapter 7 is needed for §9.4. Within this chapter there are the following dependencies:



A wide range of Lanczos papers are collected in Brown, Chu, Ellison, and Plemmons (1994). Other complementary references include Parlett (1980), Saad (1992), and Chatelin (1993). The two volume work by Cullum and Willoughby (1985a,1985b) includes both analysis and software.

9.1 Derivation and Convergence Properties

Suppose $A \in \mathbb{R}^{n\times n}$ is large, sparse, and symmetric and assume that a few of its largest and/or smallest eigenvalues are desired. This problem can be solved by a method attributed to Lanczos (1950). The method generates a sequence of tridiagonal matrices T_k with the property that the extremal eigenvalues of $T_k \in \mathbb{R}^{k\times k}$ are progressively better estimates of A's extremal eigenvalues. In this section, we derive the technique and investigate its exact arithmetic properties. Throughout the section $\lambda_i(\cdot)$ designates the ith largest eigenvalue.

9.1.1 Krylov Subspaces

The derivation of the Lanczos algorithm can proceed in several ways. So that its remarkable convergence properties do not come as a complete surprise, we prefer to lead into the technique by considering the optimization of the Rayleigh quotient

$$r(x) = \frac{x^TAx}{x^Tx} \qquad x \neq 0\,.$$

Recall from Theorem 8.1.2 that the maximum and minimum values of $r(x)$ are $\lambda_1(A)$ and $\lambda_n(A)$, respectively. Suppose $\{q_i\} \subseteq \mathbb{R}^n$ is a sequence of orthonormal vectors and define the scalars M_k and m_k by

$$M_k = \lambda_1(Q_k^TAQ_k) = \max_{y\neq 0} \frac{y^T(Q_k^TAQ_k)y}{y^Ty} = \max_{\|y\|_2=1} r(Q_ky) \leq \lambda_1(A)$$

$$m_k = \lambda_k(Q_k^T A Q_k) = \min_{y \neq 0} \frac{y^T(Q_k^T A Q_k)y}{y^T y} = \min_{\|y\|_2=1} r(Q_k y) \geq \lambda_n(A)$$

where $Q_k = [\, q_1, \ldots, q_k \,]$. The Lanczos algorithm can be derived by considering how to generate the q_k so that M_k and m_k are increasingly better estimates of $\lambda_1(A)$ and $\lambda_n(A)$.

Suppose $u_k \in \text{span}\{q_1, \ldots, q_k\}$ is such that $M_k = r(u_k)$. Since $r(x)$ increases most rapidly in the direction of the gradient

$$\nabla r(x) = \frac{2}{x^T x}(Ax - r(x)x),$$

we can ensure that $M_{k+1} > M_k$ if q_{k+1} is determined so

$$\nabla r(u_k) \in \text{span}\{q_1, \ldots, q_{k+1}\}. \tag{9.1.1}$$

(This assumes $\nabla r(u_k) \neq 0$.) Likewise, if $v_k \in \text{span}\{q_1, \ldots, q_k\}$ satisfies $r(v_k) = m_k$, then it makes sense to require

$$\nabla r(v_k) \in \text{span}\{q_1, \ldots, q_{k+1}\} \tag{9.1.2}$$

since $r(x)$ decreases most rapidly in the direction of $-\nabla r(x)$.

At first glance, the task of finding a single q_{k+1} that satisfies these two requirements appears impossible. However, since $\nabla r(x) \in \text{span}\{x, Ax\}$, it is clear that (9.1.1) and (9.1.2) can be simultaneously satisfied if

$$\text{span}\{q_1, \ldots, q_k\} = \text{span}\{q_1,\, Aq_1, \ldots, A^{k-1}q_1\}$$

and we choose q_{k+1} so

$$\text{span}\{q_1, \ldots, q_{k+1}\} = \text{span}\{q_1,\, Aq_1, \ldots, A^{k-1}q_1,\, A^k q_1\}.$$

Thus, we are led to the problem of computing orthonormal bases for the *Krylov subspaces*

$$\mathcal{K}(A, q_1, k) = \text{span}\{q_1,\, Aq_1, \ldots, A^{k-1}q_1\}.$$

These are just the range spaces of the Krylov matrices

$$K(A, q_1, n) = [\, q_1,\, Aq_1,\, A^2 q_1, \ldots,\, A^{n-1} q_1 \,].$$

presented in §8.3.2.

9.1.2 Tridiagonalization

In order to find this basis efficiently we exploit the connection between the tridiagonalization of A and the QR factorization of $K(A, q_1, n)$. Recall that if $Q^T A Q = T$ is tridiagonal with $Qe_1 = q_1$, then

$$K(A, q_1, n) = Q\, [\, e_1,\, Te_1,\, T^2 e_1, \ldots,\, T^{n-1} e_1 \,]$$

is the QR factorization of $K(A, q_1, n)$ where $e_1 = I_n(:, 1)$. Thus the q_k can effectively be generated by tridiagonalizing A with an orthogonal matrix whose first column is q_1.

Householder tridiagonalization, discussed in §8.3.1, can be adapted for this purpose. However, this approach is impractical if A is large and sparse because Householder similarity transformations tend to destroy sparsity. As a result, unacceptably large, dense matrices arise during the reduction.

Loss of sparsity can sometimes be controlled by using Givens rather than Householder transformations. See Duff and Reid (1976). However, any method that computes T by successively updating A is not useful in the majority of cases when A is sparse.

This suggests that we try to compute the elements of the tridiagonal matrix $T = Q^TAQ$ directly. Setting $Q = [\, q_1, \ldots, q_n \,]$ and

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & & \beta_{n-1} & \alpha_n \end{bmatrix}$$

and equating columns in $AQ = QT$, we find

$$Aq_k = \beta_{k-1}q_{k-1} + \alpha_k q_k + \beta_k q_{k+1} \qquad \beta_0 q_0 \equiv 0$$

for $k = 1{:}n-1$. The orthonormality of the q_i implies $\alpha_k = q_k^T A q_k$. Moreover, if $r_k = (A - \alpha_k I)q_k - \beta_{k-1}q_{k-1}$ is nonzero, then $q_{k+1} = r_k/\beta_k$ where $\beta_k = \pm\| r_k \|_2$. If $r_k = 0$, then the iteration breaks down but (as we shall see) not without the acquisition of valuable invariant subspace information. So by properly sequencing the above formulae we obtain the *Lanczos iteration*:

$$\begin{array}{l} r_0 = q_1;\ \beta_0 = 1;\ q_0 = 0;\ k = 0 \\ \textbf{while } (\beta_k \neq 0) \\ \qquad q_{k+1} = r_k/\beta_k;\ k = k+1;\ \alpha_k = q_k^T A q_k \\ \qquad r_k = (A - \alpha_k I)q_k - \beta_{k-1}q_{k-1};\ \beta_k = \| r_k \|_2 \\ \textbf{end} \end{array} \qquad (9.1.3)$$

There is no loss of generality in choosing the β_k to be positive. The q_k are called *Lanczos vectors*.

9.1.3 Termination and Error Bounds

The iteration halts before complete tridiagonalization if q_1 is contained in a proper invariant subspace. This is one of several mathematical properties of the method that we summarize in the following theorem.

Theorem 9.1.1 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and assume $q_1 \in \mathbb{R}^n$ has unit 2-norm. Then the Lanczos iteration (9.1.3) runs until $k = m$, where $m =$* $\text{rank}(K(A, q_1, n))$ *Moreover, for $k = 1{:}m$ we have*

$$AQ_k = Q_kT_k + r_ke_k^T \tag{9.1.4}$$

where

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{k-1} \\ 0 & \cdots & & \beta_{k-1} & \alpha_k \end{bmatrix}$$

and $Q_k = [\, q_1, \ldots, q_k \,]$ has orthonormal columns that span $\mathcal{K}(A, q_1, k)$.

Proof. The proof is by induction on k. Suppose the iteration has produced $Q_k = [\, q_1, \ldots, q_k \,]$ such that $\text{ran}(Q_k) = \mathcal{K}(A, q_1, k)$ and $Q_k^TQ_k = I_k$. It is easy to see from (9.1.3) that (9.1.4) holds. Thus, $Q_k^TAQ_k = T_k + Q_k^Tr_ke_k^T$. Since $\alpha_i = q_i^TAq_i$ for $i = 1{:}k$ and

$$q_{i+1}^TAq_i = q_{i+1}^T(Aq_i - \alpha_iq_i - \beta_{i-1}q_{i-1}) = q_{i+1}^T(\beta_iq_{i+1}) = \beta_i$$

for $i = 1{:}k-1$, we have $Q_k^TAQ_k = T_k$. Consequently, $Q_k^Tr_k = 0$.

If $r_k \neq 0$, then $q_{k+1} = r_k/\| r_k \|_2$ is orthogonal to $q_1, \ldots, q_k$ and

$$q_{k+1} \in \text{span}\{Aq_k,\ q_k,\ q_{k-1}\} \subseteq \mathcal{K}(A,\ q_1,\ k+1).$$

Thus, $Q_{k+1}^TQ_{k+1} = I_{k+1}$ and $\text{ran}(Q_{k+1}) = \mathcal{K}(A, q_1, k+1)$. On the other hand, if $r_k = 0$, then $AQ_k = Q_kT_k$. This says that $\text{ran}(Q_k) = \mathcal{K}(A, q_1, k)$ is invariant. From this we conclude that $k = m = \text{rank}(K(A, q_1, n))$. □

Encountering a zero β_k in the Lanczos iteration is a welcome event in that it signals the computation of an exact invariant subspace. However, an exact zero or even a small β_k is a rarity in practice. Nevertheless, the extremal eigenvalues of T_k turn out to be surprisingly good approximations to A's extremal eigenvalues. Consequently, other explanations for the convergence of T_k's eigenvalues must be sought. The following result is a step in this direction.

Theorem 9.1.2 *Suppose that k steps of the Lanczos algorithm have been performed and that $S_k^TT_kS_k = \text{diag}(\theta_1, \ldots, \theta_k)$ is the Schur decomposition of the tridiagonal matrix T_k. If $Y_k = [\, y_1, \ldots, y_k \,] = Q_kS_k \in \mathbb{R}^{n\times k}$, then for $i = 1{:}k$ we have $\| Ay_i - \theta_iy_i \|_2 = |\beta_k|\,|s_{ki}|$ where $S_k = (s_{pq})$.*

Proof. Post-multiplying (9.1.4) by S_k gives

$$AY_k = Y_k\text{diag}(\theta_1, \ldots, \theta_k) + r_ke_k^TS_k,$$

and so $Ay_i = \theta_i y_i + r_k(e_k^T S e_i)$. The proof is complete by taking norms and recalling that $\| r_k \|_2 = |\beta_k|$. □

The theorem provides computable error bounds for T_k's eigenvalues:

$$\min_{\mu \in \lambda(A)} |\theta_i - \mu| \leq |\beta_k| \, |s_{ki}| \qquad i = 1{:}k$$

Note that in the terminology of Theorem 8.1.15, the (θ_i, y_i) are Ritz pairs for the subspace $\text{ran}(Q_k)$.

Another way that T_k can be used to provide estimates of A's eigenvalues is described in Golub (1974) and involves the judicious construction of a rank-one matrix E such that $\text{ran}(Q_k)$ is invariant for $A + E$. In particular, if we use the Lanczos method to compute $AQ_k = Q_k T_k + r_k e_k^T$ and set $E = \tau w w^T$, where $\tau = \pm 1$ and $w = a q_k + b r_k$, then it can be shown that

$$(A + E)Q_k = Q_k(T_k + \tau a^2 e_k e_k^T) + (1 + \tau ab) r_k e_k^T.$$

If $0 = 1 + \tau ab$, then the eigenvalues of $\tilde{T}_k = T_k + \tau a^2 e_k e_k^T$, a tridiagonal matrix, are also eigenvalues of $A + E$. Using Theorem 8.1.8 it can be shown that the interval $[\lambda_i(\tilde{T}_k), \lambda_{i-1}(\tilde{T}_k)]$ contains an eigenvalue of A for $i = 2{:}k$.

These bracketing intervals depend on the choice of τa^2. Suppose we have an approximate eigenvalue of λ of A. One possibility is to choose τa^2 so that $\det(T_k - \lambda I_k) = (\alpha_2 + \tau a^2 - \lambda) p_{k-1}(\lambda) - \beta_{k-1}^2 p_{k-2}(\lambda) = 0$ where the polynomials $p_i(x) = \det(T_i - x I_i)$ can be evaluated at λ using the three-term recurrence (8.5.2). (This assumes that $p_{k-1}(\lambda) \neq 0$.) Eigenvalue estimation in this spirit is discussed in Lehmann (1963) and Householder (1968).

9.1.4 The Kaniel-Paige Convergence Theory

The preceding discussion indicates how eigenvalue estimates can be obtained via the Lanczos algorithm, but it reveals nothing about rate of convergence. Results of this variety constitute what is known as the *Kaniel-Paige theory*, a sample of which follows.

Theorem 9.1.3 *Let A be an n-by-n symmetric matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ and corresponding orthonormal eigenvectors $z_1, \ldots, z_n$. If $\theta_1 \geq \cdots \geq \theta_k$ are the eigenvalues of the matrix T_k obtained after k steps of the Lanczos iteration, then*

$$\lambda_1 \geq \theta_1 \geq \lambda_1 - \frac{(\lambda_1 - \lambda_n)\tan(\phi_1)^2}{(c_{k-1}(1 + 2\rho_1))^2}$$

where $\cos(\phi_1) = |q_1^T z_1|$, $\rho_1 = (\lambda_1 - \lambda_2)/(\lambda_2 - \lambda_n)$, and $c_{k-1}(x)$ is the Chebyshev polynomial of degree $k - 1$.

Proof. From Theorem 8.1.2, we have

$$\theta_1 = \max_{y\neq 0} \frac{y^T T_k y}{y^T y} = \max_{y\neq 0} \frac{(Q_k y)^T A (Q_k y)}{(Q_k y)^T (Q_k y)} = \max_{0\neq w\in\mathcal{K}(A,q_1,k)} \frac{w^T A w}{w^T w}$$

Since λ_1 is the maximum of $w^T Aw/w^T w$ over all nonzero w, it follows that $\lambda_1 \geq \theta_1$. To obtain the lower bound for θ_1, note that

$$\theta_1 = \max_{p\in\mathcal{P}_{k-1}} \frac{q_1^T p(A) A p(A) q_1}{q_1^T p(A)^2 q_1}$$

where $\mathcal{P}_{k-1}$ is the set of $k-1$ degree polynomials. If $q_1 = \sum_{i=1}^{n} d_i z_i$ then

$$\begin{aligned}\frac{q_1^T p(A) A p(A) q_1}{q_1^T p(A)^2 q_1} &= \frac{\sum_{i=1}^{n} d_i^2 p(\lambda_i)^2 \lambda_i}{\sum_{i=1}^{n} d_i^2 p(\lambda_i)^2} \\ &\geq \lambda_1 - (\lambda_1 - \lambda_n) \frac{\sum_{i=2}^{n} d_i^2 p(\lambda_i)^2}{d_1^2 p(\lambda_1)^2 + \sum_{i=2}^{n} d_i^2 p(\lambda_i)^2}.\end{aligned}$$

We can make the lower bound tight by selecting a polynomial $p(x)$ that is large at $x = \lambda_1$ in comparison to its value at the remaining eigenvalues. One way of doing this is to set

$$p(x) = c_{k-1}\left(-1 + 2\frac{x-\lambda_n}{\lambda_2 - \lambda_n}\right)$$

where $c_{k-1}(z)$ is the $(k-1)$-st Chebyshev polynomial generated via the recursion

$$c_k(z) = 2zc_{k-1}(z) - c_{k-2}(z) \qquad c_0 = 1,\ c_1 = z.$$

These polynomials are bounded by unity on $[-1,1]$, but grow very rapidly outside this interval. By defining $p(x)$ this way it follows that $|p(\lambda_i)|$ is bounded by unity for $i = 2{:}n$, while $p(\lambda_1) = c_{k-1}(1+2\rho_1)$. Thus,

$$\theta_1 \geq \lambda_1 - (\lambda_1 - \lambda_n)\frac{1-d_1^2}{d_1^2}\frac{1}{c_{k-1}(1+2\rho_1)^2}.$$

The desired lower bound is obtained by noting that $\tan(\phi_1)^2 = (1-d_1^2)/d_1^2$. □

An analogous result pertaining to θ_k follows immediately from this theorem:

Corollary 9.1.4 *Using the same notation as the theorem,*

$$\lambda_n \;\le\; \theta_k \;\le\; \lambda_n + \frac{(\lambda_1 - \lambda_n)\tan(\phi_n)^2}{c_{k-1}(1+2\rho_n)^2}$$

where $\rho_n = (\lambda_{n-1} - \lambda_n)/(\lambda_1 - \lambda_{n-1})$ *and* $\cos(\phi_n) = q_n^T z_n$.

Proof. Apply Theorem 9.1.3 with A replaced by $-A$. □

9.1.5 The Power Method Versus the Lanczos Method

It is worthwhile to compare θ_1 with the corresponding power method estimate of λ_1. (See §8.2.1.) For clarity, assume $\lambda_1 \ge \cdots \ge \lambda_n \ge 0$. After $k-1$ power method steps applied to q_1, a vector is obtained in the direction of

$$v \;=\; A^{k-1}q_1 \;=\; \sum_{i=1}^{n} c_i \lambda_i^{k-1} z_i$$

along with an eigenvalue estimate

$$\gamma_1 \;=\; \frac{v^T A v}{v^T v}.$$

Using the proof and notation of Theorem 9.1.3, it is easy to show that

$$\lambda_1 \ge \gamma_1 \ge \lambda_1 - (\lambda_1 - \lambda_n)\tan(\phi_1)^2 \left(\frac{\lambda_2}{\lambda_1}\right)^{2k-1}. \qquad (9.1.5)$$

(Hint: Set $p(x) = x^{k-1}$ in the proof.) Thus, we can compare the quality of the lower bounds for θ_1 and γ_1 by comparing

$$L_{k-1} \;\equiv\; 1\Big/\left[c_{k-1}\left(2\frac{\lambda_1}{\lambda_2} - 1\right)\right]^2 \;\ge\; 1\Big/\left[c_{k-1}(1+2\rho_1)\right]^2$$

and

$$R_{k-1} \;=\; \left(\frac{\lambda_2}{\lambda_1}\right)^{2(k-1)}.$$

This is done in following table for representative values of k and λ_2/λ_1.

The superiority of the Lanczos estimate is self-evident. This should be no surprise, since θ_1 is the maximum of $r(x) = x^T Ax/x^T x$ over *all* of $\mathcal{K}(A, q_1, k)$, while $\gamma_1 = r(v)$ for a particular v in $\mathcal{K}(A, q_1, k)$, namely $v = A^{k-1}q_1$.

λ_1/λ_2	$k = 5$	$k = 10$	$k = 15$	$k = 20$	$k = 25$
1.50	$\frac{1.1\times10^{-4}}{3.9\times10^{-2}}$	$\frac{2.0\times10^{\;10}}{6.8\times10^{-4}}$	$\frac{3.9\times10^{-16}}{1.2\times10^{-5}}$	$\frac{7.4\times10^{-22}}{2.0\times10^{-7}}$	$\frac{1.4\times10^{-27}}{3.5\times10^{-9}}$
1.10	$\frac{2.7\times10^{-2}}{4.7\times10^{-1}}$	$\frac{5.5\times10^{-5}}{1.8\times10^{-1}}$	$\frac{1.1\times10^{-7}}{6.9\times10^{-2}}$	$\frac{2.1\times10^{-10}}{2.7\times10^{-2}}$	$\frac{4.2\times10^{-13}}{1.0\times10^{-2}}$
1.01	$\frac{5.6\times10^{-1}}{9.2\times10^{-1}}$	$\frac{1.0\times10^{-1}}{8.4\times10^{-1}}$	$\frac{1.5\times10^{-2}}{7.6\times10^{-1}}$	$\frac{2.0\times10^{-3}}{6.9\times10^{-1}}$	$\frac{2.8\times10^{-4}}{6.2\times10^{-1}}$

TABLE 9.1.1 L_{k-1}/R_{k-1}

9.1.6 Convergence of Interior Eigenvalues

We conclude with some remarks about error bounds for T_k's interior eigenvalues. The key idea in the proof of Theorem 9.1.3 is the use of the translated Chebyshev polynomial. With this polynomial we amplified the component of q_1 in the direction z_1. A similar idea can be used to obtain bounds for an interior Ritz value θ_i. However, the bounds are not as satisfactory because the "amplifying polynomial" has the form $q(x)\Pi_{i=1}^{k-1}(x - \lambda_i)$, where $q(x)$ is the $(k-1)$ degree of the Chebyshev polynomial on the interval $[\lambda_{i+1}, \lambda_n]$. For details, see Kaniel (1966), Paige (1971), or Saad (1980).

Problems

P9.1.1 Suppose $A \in \mathbb{R}^{n\times n}$ is skew-symmetric. Derive a Lanczos-like algorithm for computing a skew-symmetric tridiagonal matrix T_m such that $AQ_m = Q_mT_m$, where $Q_m^TQ_m = I_m$.

P9.1.2 Let $A \in \mathbb{R}^{n\times n}$ be symmetric and define $r(x) = x^TAx/x^Tx$. Suppose $S \subseteq \mathbb{R}^n$ is a subspace with the property that $x \in S$ implies $\nabla r(x) \in S$. Show that S is invariant for A.

P9.1.3 Show that if a symmetric matrix $A \in \mathbb{R}^{n\times n}$ has a multiple eigenvalue, then the Lanczos iteration terminates prematurely.

P9.1.4 Show that the index m in Theorem 9.1.1 is the dimension of the smallest invariant subspace for A that contains q_1.

P9.1.5 Let $A \in \mathbb{R}^{n\times n}$ be symmetric and consider the problem of determining an orthonormal sequence $q_1, q_2, \ldots$ with the property that once $Q_k = [\,q_1, \ldots, q_k\,]$ is known, q_{k+1} is chosen so as to minimize $\mu_k = \| (I - Q_{k+1}Q_{k+1}^T)AQ_k \|_F$. Show that if $\text{span}\{q_1, \ldots, q_k\} = \mathcal{K}(A, q_1, k)$, then it is possible to choose q_{k+1} so $\mu_k = 0$. Explain how this optimization problem leads to the Lanczos iteration.

P9.1.6 Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and that we wish to compute its largest eigenvalue. Let η be an approximate eigenvector and set

$$\begin{aligned} \alpha &= \frac{\eta^TA\eta}{\eta^T\eta} \\ z &= A\eta - \alpha\eta. \end{aligned}$$

(a) Show that the interval $[\alpha - \delta, \alpha + \delta]$ must contain an eigenvalue of A where $\delta = \| z \|_2 / \| \eta \|_2$. (b) Consider the new approximation $\bar{\eta} = a\eta + bz$ and show how to determine the scalars a and b so that

$$\bar{\alpha} = \frac{\bar{\eta}^T A \bar{\eta}}{\bar{\eta}^T \eta}$$

is maximized. (c) Relate the above computations to the first two steps of the Lanczos process.

Notes and References for Sec. 9.1

The classic reference for the Lanczos method is

C. Lanczos (1950). "An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators," *J. Res. Nat. Bur. Stand.* *45*, 255–82.

Although the convergence of the Ritz values is alluded to this paper, for more details we refer the reader to

S. Kaniel (1966). "Estimates for Some Computational Techniques in Linear Algebra," *Math. Comp.* *20*, 369–78.

C.C. Paige (1971). "The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices," Ph.D. thesis, London University.

Y. Saad (1980). "On the Rates of Convergence of the Lanczos and the Block Lanczos Methods," *SIAM J. Num. Anal.* *17*, 687–706.

The connections between the Lanczos algorithm, orthogonal polynomials, and the theory of moments are discussed in

N.J. Lehmann (1963). "Optimale Eigenwerteinschliessungen," *Numer. Math.* *5*, 246 72.

A.S. Householder (1968). "Moments and characteristic Roots II," *Numer. Math.* *11*, 126–28.

G.H. Golub (1974). "Some Uses of the Lanczos Algorithm in Numerical Linear Algebra," *in Topics in Numerical Analysis*, ed., J.J.H. Miller, Academic Press, New York.

We motivated our discussion of the Lanczos algorithm by discussing the inevitability of fill-in when Householder or Givens transformations are used to tridiagonalize. Actually, fill-in can sometimes be kept to an acceptable level if care is exercised. See

I.S. Duff (1974). "Pivot Selection and Row Ordering in Givens Reduction on Sparse Matrices," *Computing* *13*, 239–48.

I.S. Duff and J.K. Reid (1976). "A Comparison of Some Methods for the Solution of Sparse Over-Determined Systems of Linear Equations," *J. Inst. Maths. Applic.* *17*, 267–80.

L. Kaufman (1979). "Application of Dense Householder Transformations to a Sparse Matrix," *ACM Trans. Math. Soft.* *5*, 442–50.

9.2 Practical Lanczos Procedures

Rounding errors greatly affect the behavior of the Lanczos iteration. The basic difficulty is caused by loss of orthogonality among the Lanczos vectors, a phenomenon that muddies the issue of termination and complicates the relationship between A's eigenvalues and those of the tridiagonal matrices T_k. This troublesome feature, coupled with the advent of Householder's perfectly stable method of tridiagonalization, explains why the Lanczos

algorithm was disregarded by numerical analysts during the 1950's and 1960's. However, interest in the method was rejuvenated with the development of the Kaniel-Paige theory and because the pressure to solve large, sparse eigenproblems increased with increased computer power. With many fewer than n iterations typically required to get good approximate extremal eigenvalues, the Lanczos method became attractive as a sparse matrix technique rather than as a competitor of the Householder approach.

Successful implementations of the Lanczos iteration involve much more than a simple encoding of (9.1.3). In this section we outline some of the practical ideas that have been proposed to make Lanczos procedure viable in practice.

9.2.1 Exact Arithmetic Implementation

With careful overwriting in (9.1.3) and exploitation of the formula

$$\alpha_k = q_k^T(Aq_k - \beta_{k-1}q_{k-1}),$$

the whole Lanczos process can be implemented with just two n-vectors of storage.

Algorithm 9.2.1. (The Lanczos Algorithm) Given a symmetric $A \in \mathbb{R}^{n\times n}$ and $w \in \mathbb{R}^n$ having unit 2-norm, the following algorithm computes a k-by-k symmetric tridiagonal matrix T_k with the property that $\lambda(T_k) \subset \lambda(A)$. It assumes the existence of a function **A.mult**(w) that returns the matrix-vector product Aw. The diagonal and subdiagonal elements of T_k are stored in $\alpha(1{:}k)$ and $\beta(1{:}k-1)$ respectively.

```
v(1:n) = 0; β_0 = 1; k = 0
while β_k ≠ 0
    if k ≠ 0
        for i = 1:n
            t = w_i; w_i = v_i/β_k; v_i = −β_k t
        end
    end
    v = v + A.mult(w)
    k = k + 1; α_k = w^T v; v = v − α_k w; β_k = || v ||_2
end
```

Note that A is not altered during the entire process. Only a procedure **A.mult**($\cdot$) for computing matrix-vector products involving A need be supplied. If A has an average of about i nonzeros per row, then approximately $(2i+8)n$ flops are involved in a single Lanczos step.

Upon termination the eigenvalues of T_k can be found using the symmetric tridiagonal QR algorithm or any of the special methods of §8.5, such as bisection.

The Lanczos vectors are generated in the n-vector w. If they are desired for later use, then special arrangements must be made for their storage. In the typical sparse matrix setting they could be stored on a disk or some other secondary storage device until required.

9.2.2 Roundoff Properties

The development of a practical, easy-to-use Lanczos procedure requires an appreciation of the fundamental error analyses of Paige (1971, 1976, 1980). An examination of his results is the best way to motivate the several modified Lanczos procedures of this section.

After j steps of the algorithm we obtain the matrix of computed Lanczos vectors $\hat{Q}_k = [\,\hat{q}_1, \ldots, \hat{q}_k\,]$ and the associated tridiagonal matrix

$$\hat{T}_k = \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_1 & & \cdots & 0 \\ \hat{\beta}_1 & \hat{\alpha}_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \hat{\beta}_{k-1} \\ 0 & \cdots & & \hat{\beta}_{k-1} & \hat{\alpha}_k \end{bmatrix}.$$

Paige (1971, 1976) shows that if $\hat{r}_k$ is the computed analog of r_k, then

$$A\hat{Q}_k = \hat{Q}_k\hat{T}_k + \hat{r}_k e_k^T + E_k \tag{9.2.1}$$

where

$$\| E_k \|_2 \approx \mathbf{u}\| A \|_2 . \tag{9.2.2}$$

This indicates that the important equation $AQ_k = Q_kT_k + r_ke_k^T$ is satisfied to working precision.

Unfortunately, the picture is much less rosy with respect to the orthogonality among the $\hat{q}_i$. (Normality is not an issue. The computed Lanczos vectors essentially have unit length.) If $\hat{\beta}_k = fl(\| \hat{r}_k \|_2)$ and we compute $\hat{q}_{k+1} = fl(\hat{r}_k/\hat{\beta}_k)$, then a simple analysis shows that $\hat{\beta}_k\hat{q}_{k+1} \approx \hat{r}_k + w_k$ where $\| w_k \|_2 \approx \mathbf{u}\| \hat{r}_k \|_2 \approx \mathbf{u}\| A \|_2$. Thus, we may conclude that

$$|\hat{q}_{k+1}^T\hat{q}_i| \approx \frac{|\hat{r}_k^T\hat{q}_i| + \mathbf{u}\| A \|_2}{|\hat{\beta}_k|}$$

for $i = 1{:}k$. In other words, significant departures from orthogonality can be expected when $\hat{\beta}_k$ is small, *even* in the ideal situation where $\hat{r}_k^T\hat{Q}_k$ is zero. A small $\hat{\beta}_k$ implies cancellation in the computation of $\hat{r}_k$. We stress that loss of orthogonality is due to this cancellation and is not the result of

the gradual accumulation of roundoff error.

Example 9.2.1 The matrix

$$A = \begin{bmatrix} 2.64 & -.48 \\ -.48 & 2.36 \end{bmatrix}$$

has eigenvalues $\lambda_1 = 3$ and $\lambda_2 = 2$. If the Lanczos algorithm is applied to this matrix with $q_1 = [\,.810,\ -.586\,]^T$ and three-digit floating point arithmetic is performed, then $\hat{q}_2 = [\,.707,\ .707\,]^T$. Loss of orthogonality occurs because span$\{q_1\}$ is almost invariant for A. (The vector $x = [\,.8,\ -.6\,]^T$ is the eigenvector affiliated with λ_1.)

Further details of the Paige analysis are given shortly. Suffice it to say now that loss of orthogonality always occurs in practice and with it, an apparent deterioration in the quality of $\hat{T}_k$'s eigenvalues. This can be quantified by combining (9.2.1) with Theorem 8.1.16. In particular, if in that theorem we set $F_1 = \hat{r}_k e_k^T + E_k$, $X_1 = \hat{Q}_k$, $S = \hat{T}_k$, and assume that

$$\tau = \| \hat{Q}_k^T \hat{Q}_k - I_k \|_2$$

satisfies $\tau < 1$, then there exist eigenvalues $\mu_1, \ldots, \mu_k \in \lambda(A)$ such that

$$|\mu_i - \lambda_i(T_k)| \;\le\; \sqrt{2}\,(\| \hat{r}_k \|_2 + \| E_k \|_2 + \tau(2+\tau)\| A \|_2)$$

for $i = 1{:}k$. An obvious way to control the τ factor is to orthogonalize each newly computed Lanczos vector against its predecessors. This leads directly to our first "practical" Lanczos procedure.

9.2.3 Lanczos with Complete Reorthogonalization

Let $r_0, \ldots, r_{k-1} \in \mathbb{R}^n$ be given and suppose that Householder matrices $H_0, \ldots, H_{k-1}$ have been computed such that $(H_0 \cdots H_{k-1})^T [\, r_0, \ldots, r_{k-1} \,]$ is upper triangular. Let $[\, q_1, \ldots, q_k \,]$ denote the first k columns of the Householder product $(H_0 \cdots H_{k-1})$. Now suppose that we are given a vector $r_k \in \mathbb{R}^n$ and wish to compute a unit vector q_{k+1} in the direction of

$$w \;=\; r_k - \sum_{i=1}^{k} (q_i^T r_k) q_i \;\in\; \text{span}\{q_1, \ldots, q_k\}^{\perp}.$$

If a Householder matrix H_k is determined so $(H_0 \cdots H_k)^T [\, r_0, \ldots, r_k \,]$ is upper triangular, then it follows that column $(k+1)$ of $H_0 \cdots H_k$ is the desired unit vector.

If we incorporate these Householder computations into the Lanczos process, then we can produce Lanczos vectors that are orthogonal to machine precision:

$r_0 = q_1$ (given unit vector)
Determine Householder H_0 so $H_0 r_0 = e_1$.
$\alpha_1 = q_1^T A q_1$
for $k = 1{:}n-1$
  $r_k = (A - \alpha_k I)q_k - \beta_{k-1}q_{k-1} \qquad (\beta_0 q_0 \equiv 0)$ (9.2.3)
  $w = (H_{k-1} \cdots H_0)r_k$
  Determine Householder H_k so $H_k w = (w_1, \ldots, w_k, \beta_k, 0, \ldots, 0)^T$.
  $q_{k+1} = H_0 \cdots H_k e_{k+1}$; $\alpha_{k+1} = q_{k+1}^T A q_{k+1}$
end

This is an example of a *complete reorthorgonalization* Lanczos scheme. A thorough analysis may be found in Paige (1970). The idea of using Householder matrices to enforce orthogonality appears in Golub, Underwood, and Wilkinson (1972).

That the computed $\hat{q}_i$ in (9.2.3) are orthogonal to working precision follows from the roundoff properties of Householder matrices. Note that by virtue of the definition of q_{k+1} , it makes no difference if $\beta_k = 0$. For this reason, the algorithm may safely run until $k = n-1$. (However, in practice one would terminate for a much smaller value of k.)

Of course, in any implementation of (9.2.3), one stores the Householder vectors v_k and never explicitly forms the corresponding P_k. Since we have $H_k(1{:}k, 1{:}k) = I_k$ there is no need to compute the first k components of $w = (H_{k-1} \cdots H_0)r_k$, for in exact arithmetic these components would be zero.

Unfortunately, these economies make but a small dent in the computational overhead associated with complete reorthogonalization. The Householder calculations increase the work in the kth Lanczos step by $O(kn)$ flops. Moreover, to compute q_{k+1}, the Householder vectors associated with $H_0, \ldots, H_k$ must be accessed. For large n and k, this usually implies a prohibitive amount of data transfer.

Thus, there is a high price associated with complete reorthogonalization. Fortunately, there are more effective courses of action to take, but these demand that we look more closely at how orthogonality is lost.

9.2.4 Selective Orthogonalization

A remarkable, ironic consequence of the Paige (1971) error analysis is that loss of orthogonality goes hand in hand with convergence of a Ritz pair. To be precise, suppose the symmetric QR algorithm is applied to $\hat{T}_k$ and renders computed Ritz values $\hat{\theta}_1, \ldots, \hat{\theta}_k$ and a nearly orthogonal matrix of eigenvectors $\hat{S}_k = (\hat{s}_{pq})$. If $\hat{Y}_k = [\,\hat{y}_1, \ldots, \hat{y}_k\,] = fl(\hat{Q}_k \hat{S}_k)$, then it can be shown that for $i = 1{:}k$ we have

$$|\hat{q}_{k+1}^T \hat{y}_i| \approx \frac{\mathbf{u}\| A \|_2}{|\hat{\beta}_k|\,|\hat{s}_{ki}|} \tag{9.2.4}$$

and

$$\| A\hat{y}_i - \hat{\theta}_i\hat{y}_i \|_2 \;\approx\; |\hat{\beta}_k|\,|\hat{s}_{ki}|\,. \qquad (9.2.5)$$

That is, the most recently computed Lanczos vector $\hat{q}_{k+1}$ tends to have a nontrivial and unwanted component in the direction of any converged Ritz vector. Consequently, instead of orthogonalizing $\hat{q}_{k+1}$ against all of the previously computed Lanczos vectors, we can achieve the same effect by orthogonalizing it against the much smaller set of converged Ritz vectors.

The practical aspects of enforcing orthogonality in this way are discussed in Parlett and Scott (1979). In their scheme, known as *selective orthogonalization*, a computed Ritz pair $(\hat{\theta}, \hat{y})$ is called "good" if it satisfies

$$\| A\hat{y} - \hat{\theta}\hat{y} \|_2 \;\approx\; \sqrt{\mathbf{u}}\| A \|_2\,.$$

As soon as $\hat{q}_{k+1}$ is computed, it is orthogonalized against each good Ritz vector. This is much less costly than complete reorthogonalization, since there are usually many fewer good Ritz vectors than Lanczos vectors.

One way to implement selective orthogonalization is to diagonalize $\hat{T}_k$ at each step and then examine the $\hat{s}_{ki}$ in light of (9.2.4) and (9.2.5). A much more efficient approach is to estimate the loss-of-orthogonality measure $\| I_k - \hat{Q}_k^T\hat{Q}_k \|_2$ using the following result:

Lemma 9.2.1 *Suppose* $S_+ = [\, S \;\; d \,]$ *where* $S \in \mathbb{R}^{n\times k}$ *and* $d \in \mathbb{R}^n$. *If* S *satisfies* $\| I_k - S^TS \|_2 \le \mu$ *and* $|1 - d^Td| \;\le\; \delta$ *then* $\| I_{k+1} - S_+^TS_+ \|_2 \;\le\; \mu_+$ *where*

$$\mu_+ \;=\; \frac{1}{2}\left(\mu + \delta + \sqrt{(\mu-\delta)^2 + 4\| S^Td \|_2^2}\;\right)$$

Proof. See Kahan and Parlett (1974) or Parlett and Scott (1979). □

Thus, if we have a bound for $\| I_k - \hat{Q}_k^T\hat{Q}_k \|_2$ we can generate a bound for $\| I_{k+1} - \hat{Q}_{k+1}^T\hat{Q}_{k+1} \|_2$ by applying the lemma with $S = \hat{Q}_k$ and $d = \hat{q}_{k+1}$. (In this case $\delta \approx \mathbf{u}$ and we assume that $\hat{q}_{k+1}$ has been orthogonalized against the set of currently good Ritz vectors.) It is possible to estimate the norm of $\hat{Q}_k^T\hat{q}_{k+1}$ from a simple recurrence that spares one the need for accessing $\hat{q}_1, \ldots, \hat{q}_k$. See Kahan and Parlett (1974) or Parlett and Scott (1979). The overhead is minimal, and when the bounds signal loss of orthogonality, it is time to contemplate the enlargement of the set of good Ritz vectors. Then and only then is $\hat{T}_k$ diagonalized.

9.2.5 The Ghost Eigenvalue Problem

Considerable effort has been spent in trying to develop a workable Lanczos procedure that does not involve any kind of orthogonality enforcement. Research in this direction focuses on the problem of "ghost" or "spurious"

eigenvalues. These are multiple eigenvalues of $\hat{T}_k$ that correspond to simple eigenvalues of A. They arise because the iteration essentially restarts itself when orthogonality to a converged Ritz vector is lost. (By way of analogy, consider what would happen during orthogonal iteration §8.2.8 if we "forgot" to orthogonalize.)

The problem of identifying ghost eigenvalues and coping with their presence is discussed in Cullum and Willoughby (1979) and Parlett and Reid (1981). It is a particularly pressing problem in those applications where all of A's eigenvalues are desired, for then the above orthogonalization procedures are too expensive to implement.

Difficulties with the Lanczos iteration can be expected even if A has a genuinely multiple eigenvalue. This follows because the $\hat{T}_k$ are unreduced, and unreduced tridiagonal matrices cannot have multiple eigenvalues. Our next practical Lanczos procedure attempts to circumvent this difficulty.

9.2.6 Block Lanczos

Just as the simple power method has a block analog in simultaneous iteration, so does the Lanczos algorithm have a block version. Suppose $n = rp$ and consider the decomposition

$$Q^T A Q = \bar{T} = \begin{bmatrix} M_1 & B_1^T & & \cdots & 0 \\ B_1 & M_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & B_{r-1}^T \\ 0 & \cdots & & B_{r-1} & M_r \end{bmatrix} \qquad (9.2.6)$$

where

$$Q = [X_1, \ldots, X_r] \qquad X_i \in \mathbb{R}^{n \times p}$$

is orthogonal, each $M_i \in \mathbb{R}^{p \times p}$, and each $B_i \in \mathbb{R}^{p \times p}$ is upper triangular. Comparing blocks in $AQ = Q\bar{T}$ shows that

$$AX_k = X_{k-1}B_{k-1}^T + X_k M_k + X_{k+1}B_k \qquad X_0 B_0 \equiv 0$$

for $k = 1{:}r-1$. From the orthogonality of Q we have

$$M_k = X_k^T A X_k$$

for $k = 1{:}r$. Moreover, if we define

$$R_k = AX_k - X_k M_k - X_{k-1}B_{k-1}^T \in \mathbb{R}^{n \times p}$$

then $X_{k+1}B_k = R_k$ is a QR factorization of R_k. These observations suggest that the block tridiagonal matrix $\bar{T}$ in (9.2.6) can be generated as follows:

$$
\begin{array}{l}
X_1 \in \mathbb{R}^{p\times p} \text{ given with } X_1^T X_1 = I_p. \\
M_1 = X_1^T A X_1 \\
\textbf{for } k = 1{:}r-1 \\
\quad R_k = AX_k - X_k M_k - X_{k-1}B_{k-1}^T \qquad (X_0 B_0^T \equiv 0) \\
\quad X_{k+1}B_k = R_k \qquad (\text{QR factorization of } R_k) \\
\quad M_{k+1} = X_{k+1}^T A X_{k+1} \\
\textbf{end}
\end{array}
\tag{9.2.7}
$$

At the beginning of the kth pass through the loop we have

$$
A[\, X_1, \ldots, X_k \,] \;=\; [\, X_1, \ldots, X_k \,]\bar{T}_k + R_k[\, 0, \ldots, 0, I_p \,] \tag{9.2.8}
$$

where

$$
\bar{T}_k \;=\; \begin{bmatrix} M_1 & B_1^T & & \cdots & 0 \\ B_1 & M_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & B_{k-1}^T \\ 0 & \cdots & & B_{k-1} & M_k \end{bmatrix}.
$$

Using an argument similar to the one used in the proof of Theorem 9.1.1, we can show that the X_k are mutually orthogonal provided none of the R_k are rank-deficient. However if $\text{rank}(R_k) < p$ for some k, then it is possible to choose the columns of X_{k+1} such that $X_{k+1}^T X_i = 0$, for $i = 1{:}k$. See Golub and Underwood (1977).

Because $\bar{T}_k$ has bandwidth p, it can be efficiently reduced to tridiagonal form using an algorithm of Schwartz (1968). Once tridiagonal form is achieved, the Ritz values can be obtained via the symmetric QR algorithm.

In order to intelligently decide when to use block Lanczos, it is necessary to understand how the block dimension affects convergence of the Ritz values. The following generalization of Theorem 9.1.3 sheds light on this issue.

Theorem 9.2.2 *Let A by an n-by-n symmetric matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ and corresponding orthonormal eigenvectors $z_1, \ldots, z_n$. Let $\mu_1 \geq \cdots \geq \mu_p$ be the p largest eigenvalues of the matrix $\bar{T}_k$ obtained after k steps of the block Lanczos iteration (9.2.7). If $Z_1 = [\, z_1, \ldots, z_p \,]$ and $\cos(\theta_p) = \sigma_p(Z_1^T X_1) > 0$, then for $i = 1{:}p$, $\lambda_i \;\geq\; \mu_i \;\geq\; \lambda_i - \epsilon_i^2$ where*

$$
\epsilon_i^2 \;=\; \frac{(\lambda_1 - \lambda_i)\tan^2(\theta_p)}{\left[c_{k-1}\left(\dfrac{1+\gamma_i}{1-\gamma_i}\right)\right]^2} \qquad \gamma_i \;=\; \frac{\lambda_i - \lambda_{p+1}}{\lambda_i - \lambda_n}
$$

and $c_{k-1}(z)$ *is the Chebyshev polynomial of degree* $k-1$.

Proof. See Underwood (1975). □

Analogous inequalities can be obtained for $\bar{T}_k$'s smallest eigenvalues by applying the theorem with A replaced by $-A$.

Based on Theorem 9.2.2 and scrutiny of the block Lanczos iteration (9.2.7) we may conclude that:

- the error bound for the Ritz values improve with increased p.
- the amount of work required to compute $\bar{T}_k$'s eigenvalues is proportional to p^2.
- the block dimension should be at least as large as the largest multiplicity of any sought-after eigenvalue.

How to determine block dimension in the face of these tradeoffs is discussed in detail by Scott (1979).

Loss of orthogonality also plagues the block Lanczos algorithm. However, all of the orthogonality enforcement schemes described above can be extended to the block setting.

9.2.7 s-Step Lanczos

The block Lanczos algorithm (9.2.7) can be used in an iterative fashion to calculate selected eigenalues of A. To fix ideas, suppose we wish to calculate the p largest eigenvalues. If $X_1 \in \mathbb{R}^{n\times p}$ is a given matrix having orthonormal columns, we may proceed as follows:

until $\| AX_1 - X_1\bar{T}_s \|_F$ is small enough
  Generate $X_2, \ldots, X_s \in \mathbb{R}^{n\times p}$ via the block Lanczos algorithm.
  Form $\bar{T}_s = [\, X_1, \ldots, X_s \,]^T A [\, X_1, \ldots, X_s \,]$, an sp-by-sp, p-diagonal matrix.
  Compute an orthogonal matrix $U = [\, u_1, \ldots, u_{sp} \,]$ such that $U^T\bar{T}_sU = \text{diag}(\theta_1, \ldots, \theta_{sp})$ with $\theta_1 \geq \cdots \geq \theta_{sp}$.
  Set $X_1 = [\, X_1, \ldots, X_s \,][\, u_1, \ldots, u_p \,]$.
end

This is the block analog of the *s-step Lanczos algorithm* , which has been extensively analyzed by Cullum and Donath (1974) and Underwood (1975).

The same idea can also be used to compute several of A's smallest eigenvalues or a mixture of both large and small eigenvalues. See Cullum (1978). The choice of the parameters s and p depends upon storage constraints as well as upon the factors we mentioned above in our discussion of block dimension. The block dimension p may be diminished as the good Ritz

vectors emerge. However this demands that orthogonality to the converged vectors be enforced. See Cullum and Donath (1974).

Problems

P9.2.1 Prove Lemma 9.2.1.

P9.2.2 If $\text{rank}(R_k) < p$ in (9.2.7), does it follow that $\text{range}([X_1, \ldots, X_k])$ contains an eigenvector of A?

Notes and References for Sec. 9.2

Of the several computational variants of the Lanczos Method, Algorithm 9.2.1 is the most stable. For details, see

C.C. Paige (1972). "Computational Variants of the Lanczos Method for the Eigenproblem," *J. Inst. Math. Applic. 10*, 373 81.

Other practical details associated with the implementation of the Lanczos procedure are discussed in

D.S. Scott (1979). "How to Make the Lanczos Algorithm Converge Slowly," *Math. Comp. 33*, 239–47.

B.N. Parlett, H. Simon, and L.M. Stringer (1982). "On Estimating the Largest Eigenvalue with the Lanczos Algorithm," *Math. Comp. 38*, 153–166.

B.N. Parlett and B. Nour-Omid (1985). "The Use of a Refined Error Bound When Updating Eigenvalues of Tridiagonals," *Lin. Alg. and Its Applic. 68*, 179–220.

J. Kuczyński and H. Woźniakowski (1992). "Estimating the Largest Eigenvalue by the Power and Lanczos Algorithms with a Random Start," *SIAM J. Matrix Anal. Appl. 13*, 1094–1122.

The behavior of the Lanczos method in the presence of roundoff error was originally reported in

C.C. Paige (1971). "The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices," Ph.D. thesis, University of London.

Important follow-up papers include

C.C. Paige (1976). "Error Analysis of the Lanczos Algorithm for Tridiagonalizing Symmetric Matrix," *J. Inst. Math. Applic. 18*, 341–49.

C.C. Paige (1980). "Accuracy and Effectiveness of the Lanczos Algorithm for the Symmetric Eigenproblem," *Lin. Alg. and Its Applic. 34*, 235–58.

For a discussion about various reorthogonalization schemes, see

C.C. Paige (1970). "Practical Use of the Symmetric Lanczos Process with Reorthogonalization," *BIT 10*, 183–95.

G.H. Golub, R. Underwood, and J.H. Wilkinson (1972). "The Lanczos Algorithm for the Symmetric $Ax = \lambda Bx$ Problem," Report STAN-CS-72-270, Department of Computer Science, Stanford University, Stanford, California.

B.N. Parlett and D.S. Scott (1979). "The Lanczos Algorithm with Selective Orthogonalization," *Math. Comp. 33*, 217–38.

H. Simon (1984). "Analysis of the Symmetric Lanczos Algorithm with Reorthogonalization Methods," *Lin. Alg. and Its Applic. 61*, 101–132.

Without any reorthogonalization it is necessary either to monitor the loss of orthogonality and quit at the appropriate instant or else to devise some scheme that will aid in the

distinction between the ghost eigenvalues and the actual eigenvalues. See

W. Kahan and B.N. Parlett (1976). "How Far Should You Go with the Lanczos Process?" in *Sparse Matrix Computations*, ed. J. Bunch and D. Rose, Academic Press, New York, pp. 131-44.

J. Cullum and R.A. Willoughby (1979). "Lanczos and the Computation in Specified Intervals of the Spectrum of Large, Sparse Real Symmetric Matrices, in *Sparse Matrix Proc.* , 1978, ed. I.S. Duff and G.W. Stewart, SIAM Publications, Philadelphia, PA.

B.N. Parlett and J.K. Reid (1981). "Tracking the Progress of the Lanczos Algorithm for Large Symmetric Eigenproblems," *IMA J. Num. Anal. 1,* 135–55.

D. Calvetti, L. Reichel, and D.C. Sorensen (1994). "An Implicitly Restarted Lanczos Method for Large Symmetric Eigenvalue Problems," *ETNA 2,* 1–21.

The block Lanczos algorithm is discussed in

J. Cullum and W.E. Donath (1974). "A Block Lanczos Algorithm for Computing the q Algebraically Largest Eigenvalues and a Corresponding Eigenspace of Large Sparse Real Symmetric Matrices," *Proc. of the 1974 IEEE Conf. on Decision and Control,* Phoenix, Arizona, pp. 505–9.

R. Underwood (1975). "An Iterative Block Lanczos Method for the Solution of Large Sparse Symmetric Eigenproblems," Report STAN-CS-75-495, Department of Computer Science, Stanford University, Stanford, California.

G.H. Golub and R. Underwood (1977). "The Block Lanczos Method for Computing Eigenvalues," in *Mathematical Software III* , ed. J. Rice, Academic Press, New York, pp. 364–77.

J. Cullum (1978). "The Simultaneous Computation of a Few of the Algebraically Largest and Smallest Eigenvalues of a Large Sparse Symmetric Matrix," *BIT 18,* 265–75.

A. Ruhe (1979). "Implementation Aspects of Band Lanczos Algorithms for Computation of Eigenvalues of Large Sparse Symmetric Matrices," *Math. Comp. 33,* 680-87.

The block Lanczos algorithm generates a symmetric band matrix whose eigenvalues can be computed in any of several ways. One approach is described in

H.R. Schwartz (1968). "Tridiagonalization of a Symmetric Band Matrix," *Numer. Math. 12,* 231–41. See also Wilkinson and Reinsch (1971, 273–83).

In some applications it is necessary to obtain estimates of interior eigenvalues. The Lanczos algorithm, however, tends to find the extreme eigenvalues first. The following papers deal with this issue:

A.K. Cline, G.H. Golub, and G.W. Platzman (1976). "Calculation of Normal Modes of Oceans Using a Lanczos Method," in *Sparse Matrix Computations* , ed. J.R. Bunch and D.J. Rose, Academic Press, New York, pp. 409–26.

T. Ericsson and A. Ruhe (1980). "The Spectral Transformation Lanczos Method for the Numerical Solution of Large Sparse Generalized Symmetric Eigenvalue Problems," *Math. Comp. 35,* 1251–68.

R.G. Grimes, J.G. Lewis, and H.D. Simon (1994). "A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems," *SIAM J. Matrix Anal. Appl. 15,* 228–272.

9.3 Applications to $Ax = b$ and Least Squares

In this section we briefly show how the Lanczos iteration can be embellished to solve large sparse linear equation and least squares problems. For further details, we recommend Saunders (1995).

9.3.1 Symmetric Positive Definite Systems

Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite and consider the functional $\phi(x)$ defined by

$$\phi(x) = \frac{1}{2}x^T Ax - x^T b$$

where $b \in \mathbb{R}^n$. Since $\nabla\phi(x) = Ax - b$, it follows that $x = A^{-1}b$ is the unique minimizer of ϕ. Hence, an approximate minimizer of ϕ can be regarded as an approximate solution to $Ax = b$.

Suppose $x_0 \in \mathbb{R}^n$ is an initial guess. One way to produce a vector sequence $\{x_k\}$ that converges to x is to generate a sequence of orthonormal vectors $\{q_k\}$ and to let x_k minimize ϕ over the set

$$x_0 + \text{span}\{q_1, \ldots, q_k\} \;=\; \{x_0 + a_1q_1 + \cdots + a_kq_k : a_k \in \mathbb{R}\,\}$$

for $k = 1{:}n$. If $Q_k = [\,q_1, \ldots, q_k\,]$, then this just means choosing $y \in \mathbb{R}^k$ such that

$$\begin{aligned}\phi(x_0 + Q_ky) &= \frac{1}{2}(x_0 + Q_ky)^T A(x_0 + Q_ky) - (x_0 + Q_ky)^T b \\ &= \frac{1}{2}y^T(Q_k^T AQ_k)y - y^T Q_k^T(b - Ax_0) + \phi(x_0)\end{aligned}$$

is minimized. By looking at the gradient of this expression with respect to y we see that

$$x_k \;=\; x_0 + Q_ky_k \tag{9.3.1}$$

where

$$(Q_k^T AQ_k)y_k \;=\; Q_k^T(b - Ax_0). \tag{9.3.2}$$

When $k = n$ the minimization is over all of $\mathbb{R}^n$ and so $Ax_n = b$.

For large sparse A it is necessary to overcome two hurdles in order to make this an effective solution process:

- the linear system (9.3.2) must be "easily" solved.
- we must be able to compute x_k *without* having to refer to $q_1, \ldots, q_k$ explicitly as (9.3.1) suggests. Otherwise there would be an excessive amount of data movement.

We show that both of these requirements are met if the q_k are Lanczos vectors.

After k steps of the Lanczos algorithm we obtain the factorization

$$AQ_k = Q_k T_k + r_k e_k^T \tag{9.3.3}$$

where

$$T_k = Q_k^T A Q_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{k-1} \\ 0 & \cdots & & \beta_{k-1} & \alpha_k \end{bmatrix}. \tag{9.3.4}$$

With this approach (9.3.2) becomes a symmetric positive definite tridiagonal system which may be solved via the LDL^T factorization. (See Algorithm 4.3.6.) In particular, by setting

$$L_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \mu_1 & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \mu_{k-1} & 1 \end{bmatrix} \quad \text{and} \quad D_k = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & d_k \end{bmatrix}$$

we find by comparing entries in

$$T_k = L_k D_k L_k^T \tag{9.3.5}$$

that

$d_1 = \alpha_1$
for $i = 2{:}k$
  $\mu_{i-1} = \beta_{i-1}/d_{i-1}$
  $d_i = \alpha_i - \beta_{i-1}\mu_{i-1}$
end

Note that we need only calculate the quantities

$$\begin{aligned} \mu_{k-1} &= \beta_{k-1}/d_{k-1} \\ d_k &= \alpha_k - \beta_{k-1}\mu_{k-1} \end{aligned} \tag{9.3.6}$$

in order to obtain L_k and D_k from L_{k-1} and D_{k-1}.

As we mentioned, it is critical to be able to compute x_k in (9.3.1) efficiently. To this end we define $C_k \in \mathbb{R}^{n \times k}$ and $p_k \in \mathbb{R}^k$ by the equations

$$\begin{aligned} C_k L_k^T &= Q_k \\ L_k D_k p_k &= Q_k^T(b - Ax_0) \end{aligned} \tag{9.3.7}$$

and observe that if $r_0 = b - Ax_0$ then

$$x_k \;=\; x_0 + Q_k T_k^{-1} Q_k^T r_0 \;=\; x_0 + Q_k(L_k D_k L_k^T)^{-1} Q_k^T r_0 \;=\; x_0 + C_k p_k.$$

Let $C_k = [\, c_1, \ldots, c_k \,]$ be a column partitioning. It follows from (9.3.7) that

$$[\, c_1,\; \mu_1 c_1 + c_2,\; \cdots,\; \mu_{k-1} c_{k-1} + c_k \,] \;=\; [\, q_1, \ldots, q_k \,]$$

and therefore $C_k \;=\; [\, C_{k-1}, c_k \,]$ where

$$c_k = q_k - \mu_{k-1} c_{k-1}.$$

Also observe that if we set $p_k \;=\; [\, \rho_1, \ldots, \rho_k \,]^T$ in $L_k D_k p_k \;=\; Q_k^T r_0$, then that equation becomes

$$\left[\begin{array}{ccc|c} & L_{k-1}D_{k-1} & & 0 \\ \hline 0 \;\cdots\; 0 & & \mu_{k-1}d_{k-1} & d_k \end{array}\right] \left[\begin{array}{c} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{k-1} \\ \hline \rho_k \end{array}\right] = \left[\begin{array}{c} q_1^T r_0 \\ q_2^T r_0 \\ \vdots \\ q_{k-1}^T r_0 \\ \hline q_k^T r_0 \end{array}\right].$$

Since $L_{k-1} D_{k-1} p_{k-1} = Q_{k-1}^T r_0$, it follows that

$$p_k \;=\; \left[\begin{array}{c} p_{k-1} \\ \rho_k \end{array}\right]$$

where

$$\rho_k = \left(q_k^T r_0 - \mu_{k-1} d_{k-1} \rho_{k-1} \right) / d_k$$

and thus,

$$x_k \;=\; x_0 + C_k p_k \;=\; x_0 + C_{k-1} p_{k-1} + \rho_k c_k = x_{k-1} + \rho_k c_k.$$

This is precisely the kind of recursive formula for x_k that we need. Together with (9.3.6) and (9.3.7) it enables us to make the transition from $(q_{k-1}, c_{k-1}, x_{k-1})$ to (q_k, c_k, x_k) with a minimal work and storage.

A further simplification results if we set q_1 to be a unit vector in the direction of the initial residual $r_0 = b - Ax_0$. With this choice for a Lanczos starting vector, $q_k^T r_0 = 0$ for $k \geq 2$. It follows from (9.3.3) that

$$\begin{aligned} b - Ax_k \;&=\; b - A(x_0 + Q_k y_k) \;=\; r_0 \;-\; (Q_k T_k + r_k e_k^T) y_k \\ &=\; r_0 - Q_k Q_k^T r_0 - r_k e_k^T y \;=\; -r_k e_k^T y. \end{aligned}$$

Thus, if $\beta_k = \| r_k \|_2 = 0$ in the Lanczos iteration, then $Ax_k = b$. Moreover, $\| Ax_k - b \|_2 = \beta_k |e_k^T y_k|$ and so estimates of the current residual can be obtained as a by-product of the iteration. Overall, we have the following procedure.

Algorithm 9.3.1 If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $b \in \mathbb{R}^n$, and $x_0 \in \mathbb{R}^n$ is an initial guess ($Ax_0 \approx b$), then this algorithm computes the solution to $Ax = b$.

```
r_0 = b - A x_0
β_0 = || r_0 ||_2
q_0 = 0
k = 0
while β_k ≠ 0
    q_{k+1} = r_k/β_k
    k = k + 1
    α_k = q_k^T A q_k
    r_k = (A - α_k I)q_k - β_{k-1}q_{k-1}
    β_k = || r_k ||_2
    if k = 1
        d_1 = α_1
        c_1 = q_1
        ρ_1 = β_0/α_1
        x_1 = ρ_1 q_1
    else
        μ_{k-1} = β_{k-1}/d_{k-1}
        d_k = α_k - β_{k-1}μ_{k-1}
        c_k = q_k - μ_{k-1}c_{k-1}
        ρ_k = -μ_{k-1}d_{k-1}ρ_{k-1}/d_k
        x_k = x_{k-1} + ρ_k c_k
    end
end
x = x_k
```

This algorithm requires one matrix-vector multiplication and a couple of saxpy operations per iteration. The numerical behavior of Algorithm 9.3.1 is discussed in the next chapter, where it is rederived and identified as the widely known method of *conjugate gradients.*

9.3.2 Symmetric Indefinite Systems

A key feature in the above development is the idea of computing the LDL^T factorization of the tridiagonal matrices T_k. Unfortunately, this is potentially unstable if A, and consequently T_k, is not positive definite. A way around this difficulty proposed by Paige and Saunders (1975) is to develop

the recursion for x_k via an "LQ" factorization of T_k. In particular, at the kth step of the iteration, we have Givens rotations $J_1, \ldots, J_{k-1}$ such that

$$T_k J_1 \cdots J_{k-1} \;=\; L_k \;=\; \begin{bmatrix} d_1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ e_1 & d_2 & 0 & \cdots & \cdots & \cdots & 0 \\ f_1 & e_2 & d_3 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & f_{k-2} & e_{k-1} & \bar{d}_k \end{bmatrix}.$$

Note that with this factorization x_k is given by

$$x_k \;=\; x_0 + Q_k y_k \;=\; Q_k T_k^{-1} Q_k^T b \;=\; W_k s_k$$

where

$$W_k \;=\; Q_k J_1 \cdots J_{k-1} \in \mathbb{R}^{n \times k}$$

and $s_k \in \mathbb{R}^k$ solves

$$L_k s_k \;=\; Q_k^T b.$$

Scrutiny of these equations enables one to develop a formula for computing x_k from x_{k-1} and an easily computed multiple of w_k, the last column of W_k. This defines the SYMMLQ method set forth in Paige and Saunders (1975).

A different idea is to notice from (9.3.3) and the definition $\beta_k q_{k+1} = r_k$ that

$$AQ_k = Q_k T_k + \beta_k q_{k+1} e_k^T = Q_{k+1} H_k$$

where

$$H_k = \begin{bmatrix} T_k \\ \beta_k e_k^T \end{bmatrix}.$$

This $(k+1)$-by-k matrix is upper Hessenberg and figures in the MINRES method of Paige and Saunders (1975). In this technique x_k minimizes $\| Ax - b \|_2$ over the set $x_0 + \text{span}\{q_1, \ldots, q_k\}$. Note that

$$\begin{aligned} \| A(x_0 + Q_k y) - b \|_2 \;&=\; \| AQ_k y - (b - Ax_0) \|_2 \\ &=\; \| Q_{k+1} H_k y - (b - Ax_0) \|_2 \;=\; \| H_k y - \beta_0 e_1 \|_2 \end{aligned}$$

where it is assumed that $q_1 = (b - Ax_0)/\beta_0$ is a unit vector. As in SYMMLQ, it is possible to develop recursions that permit the efficient computation of x_k from its predecessor x_{k-1}. The QR factorization of H_k is involved.

The behavior of the conjugate gradient method is detailed in the next chapter. The convergence of SYMMLQ and MINRES is more complicated and is discussed in Paige, Parlett, and Van Der Vorst (1995).

9.3.3 Bidiagonalization and the SVD

Suppose $U^TAV = B$ represents the bidiagonalization of $A \in \mathbb{R}^{m\times n}$ with

$$\begin{array}{rclcl} U & = & [\,u_1,\ldots,u_m\,] & & U^TU = I_m \\ V & = & [\,v_1,\ldots,v_n\,] & & V^TV = I_n \end{array}$$

and

$$B = \begin{bmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ 0 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & & 0 & \alpha_n \end{bmatrix}. \tag{9.3.8}$$

Recall from §5.4.3 that this factorization may be computed using Householder transformations and that it serves as a front end for the SVD algorithm.

Unfortunately, if A is large and sparse, then we can expect large, dense submatrices to arise during the Householder bidiagonalization. Consequently, it would be nice to develop a means for computing B directly without any orthogonal updates of the matrix A.

Proceeding just as we did in §9.1.2 we compare columns in the equations $AV = UB$ and $A^TU = VB^T$ for $k = 1{:}n$ and obtain

$$\begin{array}{rclcl} Av_k & = & \alpha_k u_k + \beta_{k-1}u_{k-1} & & \beta_0 u_0 \equiv 0 \\ A^Tu_k & = & \alpha_k v_k + \beta_k v_{k+1} & & \beta_n v_{n+1} \equiv 0 \end{array} \tag{9.3.9}$$

Defining

$$\begin{array}{rcl} r_k & = & Av_k - \beta_{k-1}u_{k-1} \\ p_k & = & A^Tu_k - \alpha_k v_k \end{array}$$

we may conclude from orthonormality that $\alpha_k = \pm\| r_k \|_2$, $u_k = r_k/\alpha_k$, $\beta_k = \pm\| p_k \|_2$, and $v_{k+1} = p_k/\beta_k$. Properly sequenced, these equations define the Lanczos method for bidiagonalizing a rectangular matrix:

$$\begin{array}{l} v_1 = \text{given unit 2-norm } n\text{-vector} \\ p_0 = v_1;\ \beta_0 = 1;\ k = 0;\ u_0 = 0 \\ \textbf{while } \beta_k \neq 0 \\ \quad v_{k+1} = p_k/\beta_k \\ \quad k = k+1 \\ \quad r_k = Av_k - \beta_{k-1}u_{k-1} \\ \quad \alpha_k = \| r_k \|_2 \\ \quad u_k = r_k/\alpha_k \\ \quad p_k = A^Tu_k - \alpha_k v_k \\ \quad \beta_k = \| p_k \|_2 \\ \textbf{end} \end{array} \tag{9.3.10}$$

If $\text{rank}(A) = n$, then we can guarantee that no zero α_k arise. Indeed, if $\alpha_k = 0$ then $\text{span}\{Av_1, \ldots, Av_k\} \subset \text{span}\{u_1, \ldots, u_{k-1}\}$ which implies rank deficiency.

If $\beta_k = 0$, then it is not hard to verify that

$$A\,[\,v_1, \ldots, v_k\,] \;=\; [\,u_1, \ldots, u_k\,]\,B_k$$

$$A^T\,[\,u_1, \ldots, u_k\,] \;=\; [\,v_1, \ldots, v_k\,]\,B_k^T$$

where $B_k = B(1{:}k, 1{:}k)$ and B is prescribed by (9.3.8). Thus, the v vectors and the u vectors are singular vectors and $\sigma(B_k) \subset \sigma(A)$. Lanczos bidiagonalization is discussed in Paige (1974). See also Cullum and Willoughby (1985a, 1985b). It is essentially equivalent to applying the Lanczos tridiagonalization scheme to the symmetric matrix

$$C \;=\; \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}.$$

We showed that $\lambda_i(C) \;=\; \sigma_i(A) \;=\; -\lambda_{n+m-i+1}(C)$ for $i = 1{:}n$ at the beginning of §8.6. Because of this, it is not surprising that the large singular values of the bidiagonal matrix tend to be very good approximations to the large singular values of A. The small singular values of A correspond to the interior eigenvalues of C and are not so well approximated. The equivalent of the Kaniel-Paige theory for the Lanczos bidiagonalization may be found in Luk (1978) as well as in Golub, Luk, and Overton (1981). The analytic, algorithmic, and numerical developments of the previous two sections all carry over naturally to the bidiagonalization.

9.3.4 Least Squares

The full-rank LS problem min $\|\, Ax - b\,\|_2$ can be solved via the bidiagonalization. In particular,

$$x_{LS} \;=\; Vy_{LS} \;=\; \sum_{i=1}^{n} y_i v_i$$

where $y = [y_1, \ldots, y_n]^T$ solves the system $By \;=\; [u_1^T b, \ldots, u_n^T b]^T$. Note that because B is *upper* bidiagonal, we cannot solve for y until the bidiagonalization is complete. Moreover, we are required to save the vectors $v_1, \ldots, v_n$, an unhappy circumstance if n is large.

The development of a sparse least squares algorithm based on the bidiagonalization can be accomplished more favorably if A is reduced to lower

bidiagonal form

$$U^TAV = B = \left[\begin{array}{ccccc} \alpha_1 & 0 & \cdots & \cdots & 0 \\ \beta_1 & \alpha_2 & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & & \ddots & \alpha_n \\ 0 & \cdots & \cdots & 0 & \beta_n \\ \hline & & 0 & & \end{array}\right]$$

where $V = [\, v_1, \ldots, v_n \,]$ and $U = [\, u_1, \ldots, u_m \,]$ are orthogonal. Comparing columns in the equations $A^TU = VB^T$ and $AV = UB$ we obtain

$$\begin{array}{rcl} A^Tu_k & = & \beta_{k-1}v_{k-1} + \alpha_k v_k \qquad \beta_0 v_0 \equiv 0 \\ Av_k & = & \alpha_k u_k + \beta_k u_{k+1} \end{array}.$$

It is straightforward to develop a Lanczos procedure from these equations and the resulting algorithm is very similar to (9.3.10), only u_1 is the starting vector.

Define the matrices $V_k = [\, v_1, \ldots, v_k \,]$, $U_k = [\, u_1, \ldots, u_k \,]$, and $B_k = B(1{:}k+1, 1{:}k)$ and observe that $AV_k = U_{k+1}B_k$. Our goal is to compute x_k, the minimizer of $\| \, Ax - b \, \|_2$ over all vectors of the form $x = x_0 + V_k y$, where $y \in \mathbb{R}^k$ and $x_0 \in \mathbb{R}^n$ is an initial guess. If $u_1 = (b - Ax_0)/\| \, b - Ax_0 \, \|_2$, then

$$A(x_0 + V_k y) - b = U_{k+1}B_k y - \beta_1 U_{k+1} e_1 = U_{k+1}\left(B_k y - \beta_1 e_1\right)$$

where $e_1 = I_m(:,1)$. It follows that if y_k solves the $(k+1)$-by-k lower bidiagonal LS problem

$$\min \| \, B_{k+1} y - \beta_1 e_1 \, \|_2$$

then $x_k = x_0 + V_k y_k$. Since B_k is lower bidiagonal, it is easy to compute Givens rotations $J_1, \ldots, J_k$ such that

$$J_k \cdots J_1 B_k = \left[\begin{array}{c} R_k \\ 0 \end{array}\right] \begin{array}{c} k \\ 1 \end{array}$$

is upper bidiagonal. If

$$J_k \cdots J_1 U_{k+1}^T b = \left[\begin{array}{c} d_k \\ u \end{array}\right] \begin{array}{c} k \\ 1 \end{array},$$

then it follows that $x_k = x_0 + V_k y_k = W_k d_k$ where $W_k = V_k R_k^{-1}$. Paige and Saunders (1982a) show how x_k can be obtained from x_{k-1} via a simple

recursion that involves the last column of W_k. The net result is a sparse LS algorithm referred to as LSQR that requires only a few n-vectors of storage to implement.

Problems

P9.3.1 Modify Algorithm 9.3.1 so that it implements the indefinite symmetric solver outlined in §9.3.2.

P9.3.2 How many vector workspaces are required to implement efficiently (9.3.10)?

P9.3.3 Suppose A is rank deficient and $\alpha_k = 0$ in (9.3.10). How could u_k be obtained so that the iteration could continue?

P9.3.4 Work out the lower bidiagonal version of (9.3.10) and detail the least square solver sketched in §9.3.4.

Notes and References for Sec. 9.3

Much of the material in this section has been distilled from the following papers:

C.C. Paige (1974). "Bidiagonalization of Matrices and Solution of Linear Equations," *SIAM J. Num. Anal. 11*, 197–209.

C.C. Paige and M.A. Saunders (1975). "Solution of Sparse Indefinite Systems of Linear Equations," *SIAM J. Num. Anal. 12*, 617–29.

C.C. Paige and M.A. Saunders (1982a). "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares," *ACM Trans. Math. Soft. 8*, 43–71.

C.C. Paige and M.A. Saunders (1982b). "Algorithm 583 LSQR: Sparse Linear Equations and Least Squares Problems," *ACM Trans. Math. Soft. 8*, 195–209.

M.A. Sanders (1995). "Solution of Sparse Rectangular Systems," *BIT 35*, 588–604.

See also Cullum and Willoughby (19855a,1985b) and

O. Widlund (1978). "A Lanczos Method for a Class of Nonsymmetric Systems of Linear Equations," *SIAM J. Numer. Anal. 15*, 801–12.

B.N. Parlett (1980). "A New Look at the Lanczos Algorithm for Solving Symmetric Systems of Linear Equations," *Lin. Alg. and Its Applic. 29*, 323–46.

G.H. Golub, F.T. Luk, and M. Overton (1981). "A Block Lanczos Method for Computing the Singular Values and Corresponding Singular Vectors of a Matrix," *ACM Trans. Math. Soft. 7*, 149–69.

J. Cullum, R.A. Willoughby, and M. Lake (1983). "A Lanczos Algorithm for Computing Singular Values and Vectors of Large Matrices," *SIAM J. Sci. and Stat. Comp. 4*, 197–215.

Y. Saad (1987). "On the Lanczos Method for Solving Symmetric Systems with Several Right Hand Sides," *Math. Comp. 48*, 651–662.

M. Berry and G.H. Golub (1991). "Estimating the Largest Singular Values of Large Sparse Matrices via Modified Moments," *Numerical Algorithms 1*, 353-374.

C.C. Paige, B.N. Parlett,and H.A. Van Der Vorst (1995). "Approximate Solutions and Eigenvalue Bounds from Krylov Subspaces," *Numer. Linear Algebra with Applic. 2*, 115–134.

9.4 Arnoldi and Unsymmetric Lanczos

If A is not symmetric, then the orthogonal tridiagonalization $Q^TAQ = T$ does not exist in general. There are two ways to proceed. The Arnoldi approach involves the column-by-column generation of an orthogonal Q such that $Q^TAQ = H$ is the Hessenberg reduction of §7.4. The unsymmetric Lanczos approach computes the columns of $Q = [q_1, \ldots, q_n]$ and $P = [p_1, \ldots, p_n]$ so that $P^TAQ = T$ is tridiagonal and $P^TQ = I_n$. Both methods are interesting as large sparse unsymmetric eigenvalue solvers and both can be adapted for sparse unsymmetric $Ax = b$ solving. (See §10.4.)

9.4.1 The Basic Arnoldi Iteration

One way to extend the Lanczos process to unsymmetric matrices is due to Arnoldi (1951) and revolves around the Hessenberg reduction $Q^TAQ = H$. In particular, if $Q = [\, q_1, \ldots, q_n \,]$ and we compare columns in $AQ = QH$, then

$$Aq_k = \sum_{i=1}^{k+1} h_{ik}q_i \qquad 1 \le k \le n-1 \,.$$

Isolating the last term in the summation gives

$$h_{k+1,k}q_{k+1} = Aq_k - \sum_{i=1}^{k} h_{ik}q_i :\equiv r_k$$

where $h_{ik} = q_i^TAq_k$ for $i = 1{:}k$. It follows that if $r_k \neq 0$, then q_{k+1} is specified by

$$q_{k+1} = r_k/h_{k+1,k}$$

where $h_{k+1,k} = \| \, r_k \, \|_2$. These equations define the *Arnoldi process* and in strict analogy to the symmetric Lanczos process (9.1.3) we obtain :

$$
\begin{array}{l}
r_0 = q_1 \\
h_{10} = 1 \\
k = 0 \\
\textbf{while } (h_{k+1,k} \neq 0) \\
\quad q_{k+1} = r_k/h_{k+1,k} \\
\quad k = k+1 \\
\quad r_k = Aq_k \\
\quad \textbf{for } i = 1{:}k \\
\quad\quad h_{ik} = q_i^T w \\
\quad\quad r_k = r_k - h_{ik}q_i \\
\quad \textbf{end} \\
\quad h_{k+1,k} = \| \, r_k \, \|_2 \\
\textbf{end}
\end{array}
\qquad (9.4.1)
$$

We assume that q_1 is a given unit 2-norm starting vector. The q_k are called the *Arnoldi vectors* and they define an orthonormal basis for the Krylov subspace $\mathcal{K}(A, q_1, k)$:

$$\text{span}\{q_1, \ldots, q_k\} = \text{span}\{q_1, Aq_1, \ldots, A^{k-1}q_1\}. \tag{9.4.2}$$

The situation after k steps is summarized by the *k-step Arnoldi factorization*

$$AQ_k = Q_k H_k + r_k e_k^T \tag{9.4.3}$$

where $Q_k = [q_1, \ldots, q_k]$, $e_k = I_k(:, k)$, and

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2k} \\ 0 & h_{32} & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & h_{k,k-1} & h_{kk} \end{bmatrix}.$$

If $r_k = 0$, then the columns of Q_k define an invariant subspace and $\lambda(H_k) \subseteq \lambda(A)$. Otherwise, the focus is on how to extract information about A's eigensystem from the Hessenberg matrix H_k and the matrix Q_k of Arnoldi vectors.

If $y \in \mathbb{R}^k$ is a unit 2-norm eigenvector for H_k and $H_k y = \lambda y$, then from (9.4.3)

$$(A - \lambda I)x = (e_k^T y) r_k$$

where $x = Q_k y$. We call λ a *Ritz value* and x the corresponding *Ritz vector*. The size of $|e_k^T y| \| r_k \|_2$ can be used to obtain error bounds, although the relevant perturbation theorems are not as routine to apply as in the symmetric case.

Some numerical properties of the Arnoldi iteration are discussed in Wilkinson (1965, pp.382). As with the symmetric Lanczos iteration, loss of orthogonality among the q_i is an issue. But two other features of (9.4.1) must be addressed before a practical Arnoldi eigensolver can be obtained:

- The Arnoldi vectors $q_1, \ldots, q_k$ are referenced in step k and the computation of $H_k(1{:}k, k)$ involves $O(kn)$ flops. Thus, there is a steep penalty associated with the generation of long Arnoldi sequences.

- The eigenvalues of H_k do not approximate the eigenvalues of A in the style of Kaniel and Paige. This is in contrast to the symmetric case where information about A's extremal eigenvalues emerges quickly . With Arnoldi, the early extraction of eigenvalue information depends crucially on the choice of q_1.

These realities suggest a framework in which we use Arnoldi with repeated, carefully chosen restarts and a controlled iteration maximum. (Recall the s-step Lanczos process of §9.2.7.)

9.4.2 Arnoldi with Restarting

Consider running Arnoldi for m steps and then restarting the process with a vector q_+ chosen from the span of the Arnoldi vectors $q_1, \ldots, q_m$. Because of the Krylov connection (9.4.2), q_+ has the form

$$q_+ = p(A)q_1$$

for some polynomial of degree $m-1$. If $Av_i = \lambda_i v_i$ for $i = 1{:}n$ and q_1 has the eigenvector expansion

$$q_1 = a_1 v_1 + \cdots + a_n v_n,$$

then

$$q_+ = a_1 p(\lambda_1) v_1 + \cdots + a_n p(\lambda_n) v_n.$$

Note that $\mathcal{K}(A, q_+, m)$ is rich in eigenvectors that are emphasized by $p(\lambda)$. That is, if $p(\lambda_{wanted})$ is large compared to $p(\lambda_{unwanted})$, then the Krylov space $\mathcal{K}(A, q_+, m)$ will have much better approximations to the eigenvector x_{wanted} than to the eigenvector $x_{unwanted}$. (It is possible to couch this argument in terms of Schur vectors and invariant subspaces rather than in terms of particular eigenvectors.)

Thus the act of picking a good restart vector q_+ from $\mathcal{K}(A, q_1, m)$ is the act of picking a polynomial "filter" that tunes out unwanted portions of the spectrum. Various heuristics for doing this have been developed based on computed Ritz vectors. See Saad (1980, 1984, 1992).

We describe a method due to Sorensen (1992) that determines the restart vector implicitly using the QR iteration with shifts. The restart occurs after every m steps and we assume that $m > j$ where j is the number of sought-after eigenvalues. The choice of the Arnoldi length parameter m depends on the problem dimension n, the effect of orthogonality loss, and system storage constraints.

After m steps we have the Arnoldi factorization

$$AQ_c = Q_c H_c + r_c e_m^T$$

where $Q_c \in \mathbb{R}^{n \times m}$ has orthonormal columns, $H_c \in \mathbb{R}^{m \times m}$ is upper Hessenberg, and $Q_c^T r_c = 0$. The subscript "c" stands for "current." The QR iteration with shifts is then applied to H_c:

$$
\begin{array}{l}
H^{(1)} = H_c \\
\textbf{for } i = 1{:}p \\
\quad H^{(i)} - \mu_i I = V_i R_i \\
\quad H^{(i+1)} = R_i V_i + \mu_i I \\
\textbf{end} \\
H_+ = H^{(p+1)}
\end{array}
$$

Here $p = m - j$ and it is assumed that the implicitly shifted QR process of §7.5.5 is applied. The selection of the shifts will be discussed shortly.

The orthogonal matrix $V = V_1 \cdots V_p$ has three crucial properties:

(1) $H_+ = V^T H_c V$. This is because $V_i^T H^{(i)} V_i = H^{(i+1)}$.

(2) $[V]_{mi} = 0$ for $i = 1{:}j-1$. This is because each V_i is upper Hessenberg and so $V \in \mathbb{R}^{m \times m}$ has lower bandwidth $p = m - j$.

(3) The first column of V has the form

$$Ve_1 = \alpha(H_c - \mu_p I)(H_c - \mu_{p-1} I) \cdots (H_c - \mu_1 I)e_1 \qquad (9.4.4)$$

where α is a scalar.

To be convinced of property (3), consider the $p = 2$ case:

$$\begin{aligned} VR_2R_1 &= V_1(V_2R_2)R_1 = V_1(H^{(2)} - \mu_2 I)R_1 \\ &= V_1(V_1^T H^{(1)} V_1 - \mu_2 I)R_1 = (H^{(1)} - \mu_2 I)V_1 R_1 \\ &= (H^{(1)} - \mu_2 I)(H^{(1)} - \mu_1 I) = (H_c - \mu_2 I)(H_c - \mu_1 I). \end{aligned}$$

Since R_2R_1 is upper triangular, the first column of $V = V_1V_2$ is a multiple of $(H_c - \mu_2 I)(H_c - \mu_1 I)$.

We now show how to restart the Arnoldi process using the matrix V to implicitly select the new starting vector. From (1) we obtain the following transformation of (9.4.3):

$$AQ_+ = Q_+H_+ + r_c e_m^T V$$

where $Q_+ = Q_cV$. This is *not* a new length-m Arnoldi factorization because $e_m^T V$ is not a multiple of e_m^T. However, in view of property (2),

$$AQ_+(:,1{:}j) = Q_+(:,1{:}j)H_+(1{:}j,1{:}j) + v_{mj} r_c e_j^T \qquad (9.4.5)$$

is a length-j Arnoldi factorization. By "jumping into" the basic Arnoldi iteration at step $j+1$ and performing p steps, we can extend (9.4.5) to a new length-m Arnoldi factorization. Moreover, using property (3) the associated starting vector $q_1^{(new)} = Q_+(:,1)$ has the following characterization:

$$\begin{aligned} Q_+(:,1) = Q_cVe_1 &= \alpha Q_c(H_c - \mu_p I) \cdots (H_c - \mu_1 I)e_1 \\ &= \alpha(A - \mu_p I) \cdots (A_c - \mu_1 I)Q_c e_1 \qquad (9.4.6) \end{aligned}$$

The last equation follows from the identity

$$(A - \mu I)Q_c = Q_c(H_c - \mu I) + re_m^T$$

and the fact that $e_m^T f(H_c)e_1 = 0$ for any polynomial $f(\cdot)$ of degree $p-1$ or less.

Thus, $q_1^{(new)} = p(A)q_1$ where $p(\lambda)$ is the polynomial

$$p(\lambda) = (\lambda - \mu_1)(\lambda - \mu_2)\cdots(\lambda - \mu_p).$$

This shows that the shifts are the zeros of the filtering polynomial. One interesting choice for the shifts is to compute $\lambda(H_c)$ and to identify the eigenvalues of interest $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_j$:

$$\lambda(H_c) = \{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_j\} \cup \{\tilde{\lambda}_{j+1}, \ldots, \tilde{\lambda}_m\}.$$

Setting $\mu_i = \tilde{\lambda}_{i+j}$ for $i = 1{:}p$ is one way of generating a filter polynomial that de-emphasizes the unwanted portion of the spectrum.

We have just presented the rudiments of the *implicitly restarted Arnoldi method*. It has many attractive attributes. For implementation details and further analysis, see Lehoucq and Sorensen (1996) and Morgan (1996).

9.4.3 Unsymmetric Lanczos Tridiagonalization

Another way to extend the symmetric Lanczos process is to reduce A to tridiagonal form using a general similarity transformation. Suppose $A \in \mathbb{R}^{n\times n}$ and that a nonsingular matrix Q exists so

$$Q^{-1}AQ = T = \begin{bmatrix} \alpha_1 & \gamma_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \gamma_{n-1} \\ 0 & \cdots & & \beta_{n-1} & \alpha_n \end{bmatrix}.$$

With the column partitionings

$$\begin{aligned} Q &= [\, q_1, \ldots, q_n \,] \\ Q^{-T} = P &= [\, p_1, \ldots, p_n \,] \end{aligned}$$

we find upon comparing columns in $AQ = QT$ and $A^TP = PT^T$ that

$$\begin{aligned} Aq_k &= \gamma_{k-1}q_{k-1} + \alpha_k q_k + \beta_k q_{k+1} & \gamma_0 q_0 \equiv 0 \\ A^T p_k &= \beta_{k-1}p_{k-1} + \alpha_k p_k + \gamma_k p_{k+1} & \beta_0 p_0 \equiv 0 \end{aligned}$$

for $k = 1{:}n-1$. These equations together with the *biorthogonality condition* $P^TQ = I_n$ imply

$$\alpha_k = p_k^T A q_k$$

and

$$\begin{aligned} \beta_k q_{k+1} \equiv r_k &= (A - \alpha_k I)q_k - \gamma_{k-1}q_{k-1} \\ \gamma_k p_{k+1} \equiv s_k &= (A - \alpha_k I)^T p_k - \beta_{k-1}p_{k-1}. \end{aligned}$$

There is some flexibility in choosing the scale factors β_k and γ_k. Note that

$$1 = p_{k+1}^T q_{k+1} = (s_k/\gamma_k)^T (r_k/\beta_k).$$

It follows that once β_k is specified γ_k is given by

$$\gamma_k = s_k^T r_k/\beta_k.$$

With the "canonical" choice $\beta_k = \| r_k \|_2$ we obtain

q_1, p_1 given unit 2-norm vectors with $p_1^T q_1 \neq 0$.
$k = 0$
$q_0 = 0$; $r_0 = q_1$
$p_0 = 0$; $s_0 = p_1$
while $(r_k \neq 0) \wedge (s_k \neq 0) \wedge (s_k^T r_k \neq 0)$
  $\beta_k = \| r_k \|_2$
  $\gamma_k = s_k^T r_k/\beta_k$
  $q_{k+1} = r_k/\beta_k$
  $p_{k+1} = s_k/\gamma_k$
  $k = k + 1$ (9.4.7)
  $\alpha_k = p_k^T A q_k$
  $r_k = (A - \alpha_k I)q_k - \gamma_{k-1} q_{k-1}$
  $s_k = (A - \alpha_k I)^T p_k - \beta_{k-1} p_{k-1}$
end

If

$$T_k = \begin{bmatrix} \alpha_1 & \gamma_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \gamma_{k-1} \\ 0 & \cdots & & \beta_{k-1} & \alpha_k \end{bmatrix},$$

then the situation at the bottom of the loop is summarized by the equations

$$A[\,q_1, \ldots, q_k\,] = [\,q_1, \ldots, q_k\,]T_k + r_k e_k^T \qquad (9.4.8)$$
$$A^T[\,p_1, \ldots, p_k\,] = [\,p_1, \ldots, p_k\,]T_k^T + s_k e_k^T. \qquad (9.4.9)$$

If $r_k = 0$, then the iteration terminates and span$\{q_1, \ldots, q_k\}$ is an invariant subspace for A. If $s_k = 0$, then the iteration also terminates and span$\{p_1, \ldots, p_k\}$ is an invariant subspace for A^T. However, if neither of these conditions are true and $s_k^T r_k = 0$, then the tridiagonalization process ends without any invariant subspace information. This is called *serious breakdown*. See Wilkinson (1965, p.389) for an early discussion of the matter.

9.4.4 The Look-Ahead Idea

It is interesting to look the serious breakdown issue in the block version of (9.4.7). For clarity assume that $A \in \mathbb{R}^{n\times n}$ with $n = rp$. Consider the factorization

$$P^TAQ = \begin{bmatrix} M_1 & C_1^T & & \cdots & 0 \\ B_1 & M_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & C_{r-1}^T \\ 0 & \cdots & & B_{r-1} & M_r \end{bmatrix} \qquad (9.4.10)$$

where all the blocks are p-by-p. Let $Q = [\, Q_1, \ldots, Q_r \,]$ and $P = [\, P_1, \ldots, P_r \,]$ be conformable partitionings of Q and P. Comparing block columns in the equations $AQ = QT$ and $A^TP = PT^T$ we obtain

$$\begin{aligned} Q_{k+1}B_k &= AQ_k - Q_kM_k - Q_{k-1}C_{k-1}^T &\equiv R_k \\ P_{k+1}C_k &= A^TP_k - P_kM_k^T - P_{k-1}B_{k-1}^T &\equiv S_k \end{aligned}$$

Note that $M_k = P_k^TAQ_k$. If $S_k^TR_k \in \mathbb{R}^{p\times p}$ is nonsingular and we compute $B_k, C_k \in \mathbb{R}^{p\times p}$ so that

$$C_k^TB_k = S_k^TR_k,$$

then

$$Q_{k+1} = R_kB_k^{-1} \qquad (9.4.11)$$

$$P_{k+1} = S_kC_k^{-1} \qquad (9.4.12)$$

satisfy $P_{k+1}^TQ_{k+1} = I_p$. Serious breakdown in this setting is associated with having a singular $S_k^TR_k$.

One way of solving the serious breakdown problem in (9.4.7) is to go after a factorization of the form (9.4.10) in which the block sizes are dynamically determined. Roughly speaking, in this approach matrices Q_{k+1} and P_{k+1} are built up column by column with special recursions that culminate in the production of a nonsingular $P_{k+1}^TQ_{k+1}$. The computations are arranged so that the biorthogonality conditions $P_i^TQ_{k+1} = 0$ and $Q_i^TP_{k+1} = 0$ hold for $i = 1{:}k$.

A method of this form belongs to the family of *look-ahead Lanczos* methods. The length of a look-ahead step is the width of the Q_{k+1} and P_{k+1} that it produces. If that width is one, a conventional block Lanczos step may be taken. Length-2 look-ahead steps are discussed in Parlett, Taylor and Liu (1985). The notion of *incurable breakdown* is also presented by these authors. Freund, Gutknecht, and Nachtigal (1993) cover the general case along with a host of implementation details. Floating point considerations

require the handling of "near" serious breakdown. In practice, each M_k that is 2-by-2 or larger corresponds to an instance of near serious breakdown.

Problems

P9.4.1 Prove that the Arnoldi vectors in (9.4.1) are mutually orthogonal.

P9.4.2 Prove (9.4.4).

P9.4.3 Prove (9.4.6).

P9.4.4 Give an example of a starting vector for which the unsymmetric Lanczos iteration (9.4.7) breaks down without rendering any invariant subspace information. Use

$$A = \begin{bmatrix} 1 & 6 & 2 \\ 3 & 0 & 2 \\ 1 & 3 & 5 \end{bmatrix}.$$

P9.4.5 Suppose $H \in \mathbb{R}^{n\times n}$ is upper Hessenberg. Discuss the computation of a unit upper triangular matrix U such that $HU = UT$ where T is tridiagonal.

P9.4.6 Show that the QR algorithm for eigenvalues does not preserve tridiagonal structure in the unsymmetric case.

Notes and References for Sec. 9.4

References for the Arnoldi iteration and its practical implementation include Saad (1992) and

W.E. Arnoldi (1951). "The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem," *Quarterly of Applied Mathematics 9*, 17 29.

Y. Saad (1980). "Variations of Arnoldi's Method for Computing Eigenelements of Large Unsymmetric Matrices.," *Lin. Alg. and Its Applic. 34*, 269–295.

Y. Saad (1984). "Chebyshev Acceleration Techniques for Solving Nonsymmetric Eigenvalue Problems," *Math. Comp. 42*, 567–588.

D.C. Sorensen (1992). "Implicit Application of Polynomial Filters in a k-Step Arnoldi Method," *SIAM J. Matrix Anal. Appl. 13*, 357 385.

D.C. Sorensen (1995). "Implicitly Restarted Arnoldi/Lanczos Methods for Large Scale Eigenvalue Calculations," in *Proceedings of the ICASE/LaRC Workshop on Parallel Numerical Algorithms, May 23 25, 1994*, D.E. Keyes, A. Sameh, and V. Venkatakrishnan (eds), Kluwer.

R.B, Lehoucq (1995). "Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration," Ph.D. thesis, Rice University, Houston Texas.

R.B. Lehoucq (1996). "Restarting an Arnoldi Reduction," Report MCS-P591-0496, Argonne National Laboratory, Argonne Illinois.

R.B. Lehoucq and D.C. Sorensen (1996). "Deflation Techniques for an Implicitly Restarted Iteration," *SIAM J. Matrix Analysis and Applic*, to appear.

R.B. Morgan (1996). "On Restarting the Arnoldi Method for Large Nonsymmetric Eigenvalue Problems," *Math Comp 65*, 1213–1230.

Related papers include

A. Ruhe (1984). "Rational Krylov Algorithms for Eigenvalue Computation," *Lin. Alg. and Its Applic. 58*, 391 405.

A. Ruhe (1994). "Rational Krylov Algorithms for Nonsymmetric Eigenvalue Problems II. Matrix Pairs," *Lin. Alg. and Its Applic. 197*, 283–295.

A. Ruhe (1994). "The Rational Krylov Algorithm for Nonsymmetric Eigenvalue Problems III: Complex Shifts for Real Matrices," *BIT 34*,165–176.

T. Huckle (1994). "The Arnoldi Method for Normal Matrices," *SIAM J. Matrix Anal. Appl. 15*, 479–489.

C.C. Paige, B.N. Parlett,and H.A. Van Der Vorst (1995). "Approximate Solutions and Eigenvalue Bounds from Krylov Subspaces," *Numer. Linear Algebra with Applic. 2*, 115–134.

K.C. Toh and L.N. Trefethen (1996). "Calculation of Pseudospectra by the Arnoldi Iteration," *SIAM J. Sci. Comp. 17*, 1–15.

The unsymmetric Lanczos process and related look ahead ideas are nicely presented in

B.N. Parlett, D. Taylor, and Z. Liu (1985). "A Look-Ahead Lanczos Algorithm for Unsymmetric Matrices," *Math. Comp. 44*, 105–124.

R.W. Freund, M. Gutknecht, and N. Nachtigal (1993). "An Implementation of the Look-Ahead Lanczos Algorithm for Non-Hermitian Matrices," *SIAM J. Sci. and Stat.Comp. 14*, 137–158.

See also

Y. Saad (1982). "The Lanczos Biorthogonalization Algorithm and Other Oblique Projection Methods for Solving Large Unsymmetric Eigenproblems," *SIAM J. Numer. Anal. 19*, 485–506.

G.A. Geist (1991). "Reduction of a General Matrix to Tridiagonal Form," *SIAM J. Matrix Anal. Appl. 12*, 362–373.

C. Brezinski, M. Zaglia, and H. Sadok (1991). "Avoiding Breakdown and Near Breakdown in Lanczos Tyoe Algorithms," *Numer. Alg. 1*, 261–284.

S.K. Kim and A.T. Chronopoulos (1991). "A Class of Lanczos-Like Algorithms Implemented on Parallel Computers," *Parallel Comput. 17*, 763–778.

B.N. Parlett (1992). "Reduction to Tridiagonal Form and Minimal Realizations," *SIAM J. Matrix Anal. Appl. 13*, 567–593.

M. Gutknecht (1992). "A Completed Theory of the Unsymmetric Lanczos Process and Related Algorithms, Part I," *SIAM J. Matrix Anal. Appl. 13*, 594–639.

M. Gutknecht (1994). "A Completed Theory of the Unsymmetric Lanczos Process and Related Algorithms, Part II," *SIAM J. Matrix Anal. Appl. 15*, 15–58.

Z. Bai (1994). "Error Analysis of the Lanczos Algorithm for Nonsymmetric Eigenvalue Problem," *Math. Comp. 62*, 209–226.

T. Huckle (1995). "Low-Rank Modification of the Unsymmetric Lanczos Algorithm," *Math.Comp. 64*, 1577–1588.

Z. Jia (1995). "The Convergence of Generalized Lanczos Methods for Large Unsymmetric Eigenproblems," *SIAM J. Matrix Anal. Applic 16*, 543–562.

M.T. Chu, R.E. Funderlic, and G.H. Golub (1995). "A Rank-One Reduction Formula and Its Applications to Matrix Factorizations," *SIAM Review 37*, 512–530.

Other papers include

H.A. Van der Vorst (1982). "A Generalized Lanczos Scheme," *Math. Comp. 39*, 559–562.

D. Boley and G.H. Golub (1984). "The Lanczos-Arnoldi Algorithm and Controllability," *Syst. Control Lett. 4*, 317–324.

Chapter 10

Iterative Methods for Linear Systems

We concluded the previous chapter by showing how the Lanczos iteration could be used to solve various linear equation and least squares problems. The methods developed were suitable for large sparse problems because they did not require the factorization of the underlying matrix. In this section, we continue the discussion of linear equation solvers that have this property.

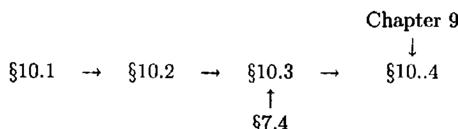
The first section is a brisk review of the classical iterations: Jacobi, Gauss-Seidel, SOR, Chebyshev semi-iterative, and so on. Our treatment of these methods is brief because our principal aim in this chapter is to highlight the method of conjugate gradients. In §10.2, we carefully develop this important technique in a natural way from the method of steepest descent. Recall that the conjugate gradient method has already been introduced via the Lanczos iteration in §9.3. The reason for deriving the method again is to motivate some of its practical variants, which are the subject of §10.3. Extensions to unsymmetric problems are treated in §10.4.

We warn the reader of an inconsistency in the notation of this chapter In §10.1, methods are developed at the "(i, j) level" necessitating the use of superscripts: $x_i^{(k)}$ denotes the i-th component of a vector $x^{(k)}$. In the other

sections, however, algorithmic developments can proceed without explicit mention of vector/matrix entries. Hence, in §10.2-§10.4 we dispense with superscripts and denote vector sequences by $\{x_k\}$.

Before You Begin

Chapter 1, §§2.1-2.5, and §2.7, Chapter 3, and §§4.1-4.3 are assumed. Other dependencies include:



Texts devoted to iterative solvers include Varga (1962), Young (1971), Hageman and Young (1981), and Axelsson (1994). The software "templates" volume by Barrett *et al* (1993) is particularly useful. The direct (non-iterative) solution of large sparse systems is sometimes preferred. See George and Liu (1981) and Duff, Erisman, and Reid (1986).

10.1 The Standard Iterations

The linear equation solvers in Chapters 3 and 4 involve the factorization of the coefficient matrix A. Methods of this type are called *direct methods.* Direct methods can be impractical if A is large and sparse, because the sought-after factors can be dense. An exception to this occurs when A is banded (cf. §4.3). Yet in many band matrix problems even the band itself is sparse making algorithms such as band Cholesky difficult to implement.

One reason for the great interest in sparse linear equation solvers is the importance of being able to obtain numerical solutions to partial differential equations. Indeed, researchers in computational PDE's have been responsible for many of the sparse matrix techniques that are presently in general use.

Roughly speaking, there are two approaches to the sparse $Ax = b$ problem. One is to pick an appropriate direct method and adapt it to exploit A's sparsity. Typical adaptation strategies involve the intelligent use of data structures and special pivoting strategies that minimize fill-in.

In contrast to the direct methods are the *iterative methods.* These methods generate a sequence of approximate solutions $\{x^{(k)}\}$ and essentially involve the matrix A only in the context of matrix-vector multiplication. The evaluation of an iterative method invariably focuses on how quickly the iterates $x^{(k)}$ converge. In this section, we present some basic iterative methods, discuss their practical implementation, and prove a few representative theorems concerned with their behavior.

10.1.1 The Jacobi and Gauss-Seidel Iterations

Perhaps the simplest iterative scheme is the *Jacobi iteration*. It is defined for matrices that have nonzero diagonal elements. The method can be motivated by rewriting the 3-by-3 system $Ax = b$ as follows:

$$\begin{aligned} x_1 &= (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11} \\ x_2 &= (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22} \\ x_3 &= (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \end{aligned}$$

Suppose $x^{(k)}$ is an approximation to $x = A^{-1}b$. A natural way to generate a new approximation $x^{(k+1)}$ is to compute

$$\begin{aligned} x_1^{(k+1)} &= (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)})/a_{11} \\ x_2^{(k+1)} &= (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)})/a_{22} \\ x_3^{(k+1)} &= (b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)})/a_{33} \end{aligned} \tag{10.1.1}$$

This defines the Jacobi iteration for the case $n = 3$. For general n we have

for $i = 1{:}n$

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) \Big/ a_{ii} \tag{10.1.2}$$

end

Note that in the Jacobi iteration one does not use the most recently available information when computing $x_i^{(k+1)}$. For example, $x_1^{(k)}$ is used in the calculation of $x_2^{(k+1)}$ even though component $x_1^{(k+1)}$ is known. If we revise the Jacobi iteration so that we always use the most current estimate of the exact x_i then we obtain

for $i = 1{:}n$

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) \Big/ a_{ii} \tag{10.1.3}$$

end

This defines what is called the *Gauss-Seidel iteration*.

For both the Jacobi and Gauss-Seidel iterations, the transition from $x^{(k)}$ to $x^{(k+1)}$ can be succinctly described in terms of the matrices L, D, and U defined by:

$$L = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ a_{21} & 0 & \cdots & & \vdots \\ a_{31} & a_{32} & \ddots & & 0 \\ \vdots & & & 0 & 0 \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & 0 \end{bmatrix}$$

$$D = \text{diag}(a_{11}, \ldots, a_{nn}) \tag{10.1.4}$$

$$U = \begin{bmatrix} 0 & a_{12} & \cdots & \cdots & a_{1n} \\ 0 & 0 & \cdots & & \vdots \\ 0 & 0 & \ddots & & a_{n-2,n} \\ \vdots & & & & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

In particular, the Jacobi step has the form $M_J x^{(k+1)} = N_J x^{(k)} + b$ where $M_J = D$ and $N_J = -(L+U)$. On the other hand, Gauss-Seidel is defined by $M_G x^{(k+1)} = N_G x^{(k)} + b$ with $M_G = (D+L)$ and $N_G = -U$.

10.1.2 Splittings and Convergence

The Jacobi and Gauss-Seidel procedures are typical members of a large family of iterations that have the form

$$Mx^{(k+1)} = Nx^{(k)} + b \tag{10.1.5}$$

where $A = M - N$ is a *splitting* of the matrix A. For the iteration (10.1.5) to be practical, it must be "easy" to solve a linear system with M as the matrix. Note that for Jacobi and Gauss-Seidel, M is diagonal and lower triangular respectively.

Whether or not (10.1.5) converges to $x = A^{-1}b$ depends upon the eigenvalues of $M^{-1}N$. In particular, if the *spectral radius* of an n-by-n matrix G is defined by

$$\rho(G) = \max\{\, |\lambda| : \lambda \in \lambda(G)\,\},$$

then it is the size of $\rho(M^{-1}N)$ is critical to the success of (10.1.5).

Theorem 10.1.1 *Suppose $b \in \mathbb{R}^n$ and $A = M - N \in \mathbb{R}^{n\times n}$ is nonsingular. If M is nonsingular and the spectral radius of $M^{-1}N$ satisfies the inequality $\rho(M^{-1}N) < 1$, then the iterates $x^{(k)}$ defined by $Mx^{(k+1)} = Nx^{(k)} + b$ converge to $x = A^{-1}b$ for any starting vector $x^{(0)}$.*

Proof. Let $e^{(k)} = x^{(k)} - x$ denote the error in the kth iterate. Since $Mx = Nx + b$ it follows that $M(x^{(k+1)} - x) = N(x^{(k)} - x)$, and thus, the error in $x^{(k+1)}$ is given by $e^{(k+1)} = M^{-1}Ne^{(k)} = (M^{-1}N)^{k+1}e^{(0)}$. From Lemma 7.3.2 we know that $(M^{-1}N)^k \to 0$ iff $\rho(M^{-1}N) < 1$. □

This result is central to the study of iterative methods where algorithmic development typically proceeds along the following lines:

- A splitting $A = M - N$ is proposed where linear systems of the form $Mz = d$ are "easy" to solve.

- Classes of matrices are identified for which the iteration matrix $G = M^{-1}N$ satisfies $\rho(G) < 1$.
- Further results about $\rho(G)$ are established to gain intuition about how the error $e^{(k)}$ tends to zero.

For example, consider the Jacobi iteration, $Dx^{(k+1)} = -(L+U)x^{(k)} + b$. One condition that guarantees $\rho(M_J^{-1}N_J) < 1$ is strict diagonal dominance. Indeed, if A has that property (defined in §3.4.10), then

$$\rho(M_J^{-1}N_J) \le \| D^{-1}(L+U) \|_\infty = \max_{1\le i\le n} \sum_{\substack{j=1 \\ j\ne i}}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| < 1$$

Usually, the "more dominant" the diagonal the more rapid the convergence but there are counterexamples. See P10.1.7.

A more complicated spectral radius argument is needed to show that Gauss-Seidel converges for symmetric positive definite A.

Theorem 10.1.2 *If $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite, then the Gauss-Seidel iteration (10.1.3) converges for any $x^{(0)}$.*

Proof. Write $A = L + D + L^T$ where $D = \text{diag}(a_{ii})$ and L is strictly lower triangular. In light of Theorem 10.1.1 our task is to show that the matrix $G = -(D+L)^{-1}L^T$ has eigenvalues that are inside the unit circle. Since D is positive definite we have $G_1 \equiv D^{1/2}GD^{-1/2} = -(I+L_1)^{-1}L_1^T$, where $L_1 = D^{-1/2}LD^{-1/2}$. Since G and G_1 have the same eigenvalues, we must verify that $\rho(G_1) < 1$. If $G_1x = \lambda x$ with $x^Hx = 1$, then we have $-L_1^Tx = \lambda(I+L_1)x$ and thus, $-x^HL_1^Tx = \lambda(1 + x^HL_1x)$. Letting $a + bi = x^HL_1x$ we have

$$|\lambda|^2 = \left| \frac{-a+bi}{1+a+bi} \right|^2 = \frac{a^2+b^2}{1+2a+a^2+b^2}.$$

However, since $D^{-1/2}AD^{-1/2} = I + L_1 + L_1^T$ is positive definite, it is not hard to show that $0 < 1 + x^HL_1x + x^HL_1^Tx = 1 + 2a$ implying $|\lambda| < 1$. □

This result is frequently applicable because many of the matrices that arise from discretized elliptic PDE's are symmetric positive definite. Numerous other results of this flavor appear in the literature.

10.1.3 Practical Implementation of Gauss-Seidel

We now focus on some practical details associated with the Gauss-Seidel iteration. With overwriting the Gauss-Seidel step (10.1.3) is particularly simple to implement:

for $i = 1{:}n$

$$x_i = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}x_j \right) \Big/ a_{ii}$$

end

This computation requires about twice as many flops as there are nonzero entries in the matrix A. It makes no sense to be more precise about the work involved because the actual implementation depends greatly upon the structure of the problem at hand.

In order to stress this point we consider the application of (10.1.3) to the NM-by-NM block tridiagonal system

$$\begin{bmatrix} T & -I_N & & \cdots & 0 \\ -I_N & T & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & -I_N \\ 0 & \cdots & & -I_N & T \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ \vdots \\ g_M \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_M \end{bmatrix} \tag{10.1.6}$$

where

$$T = \begin{bmatrix} 4 & -1 & & \cdots & 0 \\ -1 & 4 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & -1 \\ 0 & \cdots & & -1 & 4 \end{bmatrix}, \quad g_j = \begin{bmatrix} G(1,j) \\ G(2,j) \\ \vdots \\ \vdots \\ G(N,j) \end{bmatrix}, \quad f_j = \begin{bmatrix} F(1,j) \\ F(2,j) \\ \vdots \\ \vdots \\ F(N,j) \end{bmatrix}.$$

This problem arises when the Poisson equation is discretized on a rectangle. It is easy to show that the matrix A is positive definite.

With the convention that $G(i,j) = 0$ whenever $i \in \{0, N+1\}$ or $j \in \{0, M+1\}$ we see that with overwriting the Gauss-Seidel step takes on the form:

for $j = 1{:}M$
  for $i = 1{:}N$
    $G(i,j) = (F(i,j) + G(i-1,j) + G(i+1,j) +$
        $G(i,j-1) + G(i,j+1))/4$
  end
end

Note that in this problem no storage is required for the matrix A.

10.1.4 Successive Over-Relaxation

The Gauss-Seidel iteration is very attractive because of its simplicity. Unfortunately, if the spectral radius of $M_G^{-1}N_G$ is close to unity, then it may be prohibitively slow because the error tends to zero like $\rho(M_G^{-1}N_G)^k$. To rectify this, let $\omega \in \mathbb{R}$ and consider the following modification of the Gauss-Seidel step:

for $i = 1{:}n$

$$x_i^{(k+1)} = \omega\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) \Big/ a_{ii} + (1-\omega)x_i^{(k)} \qquad (10.1.7)$$

end

This defines the method of *successive over-relaxation* (SOR). Using (10.1.4) we see that in matrix terms, the SOR step is given by

$$M_\omega x^{(k+1)} = N_\omega x^{(k)} + \omega b \qquad (10.1.8)$$

where $M_\omega = D + \omega L$ and $N_\omega = (1-\omega)D - \omega U$. For a few structured (but important) problems such as (10.1.6), the value of the *relaxation parameter* ω that minimizes $\rho(M_\omega^{-1}N_\omega)$ is known. Moreover, a significant reduction in $\rho(M_1^{-1}N_1) = \rho(M_G^{-1}N_G)$ can result. In more complicated problems, however, it may be necessary to perform a fairly sophisticated eigenvalue analysis in order to determine an appropriate ω.

10.1.5 The Chebyshev Semi-Iterative Method

Another way to accelerate the convergence of an iterative method makes use of Chebyshev polynomials. Suppose $x^{(1)}, \ldots, x^{(k)}$ have been generated via the iteration $Mx^{(j+1)} = Nx^{(j)} + b$ and that we wish to determine coefficients $\nu_j(k)$, $j = 0{:}k$ such that

$$y^{(k)} = \sum_{j=0}^{k} \nu_j(k)x^{(j)} \qquad (10.1.9)$$

represents an improvement over $x^{(k)}$. If $x^{(0)} = \cdots = x^{(k)} = x$, then it is reasonable to insist that $y^{(k)} = x$. Hence, we require

$$\sum_{j=0}^{k} \nu_j(k) = 1\,. \qquad (10.1.10)$$

Subject to this constraint, we consider how to choose the $\nu_j(k)$ so that the error in $y^{(k)}$ is minimized.

Recalling from the proof of Theorem 10.1.1 that $x^{(k)} - x = (M^{-1}N)^k e^{(0)}$ where $e^{(0)} = x^{(0)} - x$, we see that

$$y^{(k)} - x \;=\; \sum_{j=0}^{k} \nu_j(k)(x^{(j)} - x) \;=\; \sum_{j=0}^{k} \nu_j(k)(M^{-1}N)^j e^{(0)} .$$

Working in the 2-norm we therefore obtain

$$\| \, y^{(k)} - x \, \|_2 \;\le\; \| \, p_k(G) \, \|_2 \, \| \, e^{(0)} \, \|_2 \qquad (10.1.11)$$

where $G = M^{-1}N$ and

$$p_k(z) \;=\; \sum_{j=0}^{k} \nu_j(k) z^j .$$

Note that the condition (10.1.10) implies $p_k(1) = 1$.

At this point we assume that G is symmetric with eigenvalues λ_i that satisfy $-1 < \alpha \le \lambda_n \le \cdots \le \lambda_1 \le \beta < 1$. It follows that

$$\| \, p_k(G) \, \|_2 \;=\; \max_{\lambda_i \in \lambda(A)} |p_k(\lambda_i)| \;\le\; \max_{\alpha \le \lambda \le \beta} |p_k(\lambda)| .$$

Thus, to make the norm of $p_k(G)$ small, we need a polynomial $p_k(z)$ that is small on $[\alpha, \beta]$ subject to the constraint that $p_k(1) = 1$.

Consider the Chebyshev polynomials $c_j(z)$ generated by the recursion $c_j(z) = 2zc_{j-1}(z) - c_{j-2}(z)$ where $c_0(z) = 1$ and $c_1(z) = z$. These polynomials satisfy $|c_j(z)| \le 1$ on [-1, 1] but grow rapidly off this interval. As a consequence, the polynomial

$$p_k(z) \;=\; \frac{c_k\left(-1 + 2\dfrac{z - \alpha}{\beta - \alpha}\right)}{c_k(\mu)}$$

where

$$\mu \;=\; -1 + 2\frac{1 - \alpha}{\beta - \alpha} \;=\; 1 + 2\frac{1 - \beta}{\beta - \alpha}$$

satisfies $p_k(1) = 1$ and tends to be small on $[\alpha, \beta]$. From the definition of $p_k(z)$ and equation (10.1.11) we see

$$\| \, y^{(k)} - x \, \|_2 \;\le\; \frac{\| \, x - x^{(0)} \, \|_2}{|c_k(\mu)|} .$$

Thus, the larger μ is, the greater the acceleration of convergence.

In order for the above to be a practical acceleration procedure, we need a more efficient method for calculating $y^{(k)}$ than (10.1.9). We have been

tacitly assuming that n is large and thus the retrieval of $x^{(0)}, \ldots, x^{(k)}$ for large k would be inconvenient or even impossible.

Fortunately, it is possible to derive a three-term recurrence among the $y^{(k)}$ by exploiting the three-term recurrence among the Chebyshev polynomials. In particular, it can be shown that if

$$\omega_{k+1} = 2\frac{2-\beta-\alpha}{\beta-\alpha}\frac{c_k(\mu)}{c_{k+1}(\mu)}$$

then

$$\begin{aligned} y^{(k+1)} &= \omega_{k+1}(y^{(k)} - y^{(k-1)} + \gamma z^{(k)}) + y^{(k-1)} \\ Mz^{(k)} &= b - Ay^{(k)} \\ \gamma &= 2/(2-\alpha-\beta), \end{aligned} \tag{10.1.12}$$

where $y^{(0)} = x^{(0)}$ and $y^{(1)} = x^{(1)}$. We refer to this scheme as the Chebyshev semi-iterative method associated with $My^{(k+1)} = Ny^{(k)} + b$. For the acceleration to be effective we need good lower and upper bounds α and β. As in SOR, these parameters may be difficult to ascertain except in a few structured problems.

Chebyshev semi-iterative methods are extensively analyzed in Varga (1962, chapter 5), as well as in Golub and Varga (1961).

10.1.6 Symmetric SOR

In deriving the Chebyshev acceleration we assumed that the iteration matrix $G = M^{-1}N$ was symmetric. Thus, our simple analysis does not apply to the unsymmetric SOR iteration matrix $M_\omega^{-1}N_\omega$. However, it is possible to symmetrize the SOR method making it amenable to Chebyshev acceleration. The idea is to couple SOR with the *backward SOR* scheme

for $i = n: -1:1$

$$x_i^{(k+1)} = \omega\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)\Big/ a_{ii} + (1-\omega)x_i^{(k)} \tag{10.1.13}$$

end

This iteration is obtained by updating the unknowns in reverse order in (10.1.7). Backward SOR can be described in matrix terms using (10.1.4). In particular, we have $\tilde{M}_\omega x^{(k+1)} = \tilde{N}_\omega x^{(k)} + \omega b$ where

$$\tilde{M}_\omega = D + \omega U \qquad \text{and} \qquad \tilde{N}_\omega = (1-\omega)D - \omega L. \tag{10.1.14}$$

If A is symmetric ($U = L^T$), then $\tilde{M}_\omega = M_\omega^T$ and $\tilde{N}_\omega = N_\omega^T$, and we have the iteration

$$\begin{aligned} M_\omega x^{(k+1/2)} &= N_\omega x^{(k)} + \omega b \\ M_\omega^T x^{(k+1)} &= N_\omega^T x^{(k+1/2)} + \omega b. \end{aligned} \tag{10.1.15}$$

It is clear that $G = M_\omega^{-T} N_\omega^T M_\omega^{-1} N_\omega$ is the iteration matrix for this method. From the definitions of M_ω and N_ω it follows that

$$G = M^{-1}N \equiv (M_\omega D^{-1} M_\omega^T)^{-1}(N_\omega^T D^{-1} N_\omega). \tag{10.1.16}$$

If D has positive diagonal entries and $KK^T = (N_\omega^T D^{-1} N_\omega)$ is the Cholesky factorization, then $K^T G K^{-T} = K^T(M_\omega D^{-1} M_\omega^T)^{-1} K$. Thus, G is similar to a symmetric matrix and has real eigenvalues.

The iteration (10.1.15) is called the *symmetric successive over-relaxation* (SSOR) method. It is frequently used in conjunction with the Chebyshev semi-iterative acceleration.

Problems

P10.1.1 Show that the Jacobi iteration can be written in the form $x^{(k+1)} = x^{(k)} + Hr^{(k)}$ where $r^{(k)} = b - Ax^{(k)}$. Repeat for the Gauss-Seidel iteration.

P10.1.2 Show that if A is strictly diagonally dominant, then the Gauss-Seidel iteration converges.

P10.1.3 Show that the Jacobi iteration converges for 2-by-2 symmetric positive definite systems.

P10.1.4 Show that if $A = M - N$ is singular, then we can never have $\rho(M^{-1}N) < 1$ even if M is nonsingular.

P10.1.5 Prove (10.1.16).

P10.1.6 Prove the converse of Theorem 10.1.1. In other words, show that if the iteration $Mx^{(k+1)} = Nx^{(k)} + b$ always converges, then $\rho(M^{-1}N) < 1$.

P10.1.7 (Supplied by R.S. Varga) Suppose that

$$A_1 = \begin{bmatrix} 1 & -1/2 \\ -1/2 & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 1 & -3/4 \\ -1/12 & 1 \end{bmatrix}.$$

Let J_1 and J_2 be the associated Jacobi iteration matrices. Show that $\rho(J_1) > \rho(J_2)$ thereby refuting the claim that greater diagonal dominance implies more rapid Jacobi convergence.

P10.1.8 The Chebyshev algorithm is defined in terms of parameters

$$\omega_{k+1} = \frac{2c_k(1/\rho)}{\rho c_{k+1}(1/\rho)}$$

where $c_k(\lambda) = \cosh[k\cosh^{-1}(\lambda)]$ with $\lambda > 1$. (a) Show that $1 < \omega_k < 2$ for $k > 1$ whenever $0 < \rho < 1$. (b) Verify that $\omega_{k+1} < \omega_k$. (c) Determine $\lim \omega_k$ as $k \to \infty$.

P10.1.9 Consider the 2-by-2 matrix

$$A = \begin{bmatrix} 1 & \rho \\ -\rho & 1 \end{bmatrix}.$$

(a) Under what conditions will Gauss-Seidel converge with this matrix? (b) For what range of ω will the SOR method converge? What is the optimal choice for this parameter? (c) Repeat (a) and (b) for the matrix

$$A = \begin{bmatrix} I_n & S \\ -S^T & I_n \end{bmatrix}$$

where $S \in \mathbb{R}^{n \times n}$. Hint: Use the SVD of S.

P10.1.10 We want to investigate the solution of $Au = f$ where $A \neq A^T$. For a model problem, consider the finite difference approximation to

$$-u'' + \sigma u' = 0 \qquad 0 < x < 1$$

where $u(0) = 10$ and $u(1) = 10\exp^{\sigma}$. This leads to the difference equation

$$-u_{i-1} + 2u_i - u_{i+1} + R(u_{i+1} - u_{i-1}) = 0 \qquad i = 1{:}n$$

where $R = \sigma h/2$, $u_0 = 10$, and $u_{n+1} = 10\exp^{\sigma}$. The number R should be less than 1. What is the convergence rate for the iteration $Mu^{(k+1)} = Nu^{(k)} + f$ where $M = (A + A^T)/2$ and $N = (A^T - A)/2$?

P10.1.11 Consider the iteration

$$y^{(k+1)} = \omega(By^{(k)} + d - y^{(k-1)}) + y^{(k-1)}$$

where B has Schur decomposition $Q^TBQ = \text{diag}(\lambda_1, \ldots, \lambda_n)$ with $\lambda_1 \geq \cdots \geq \lambda_n$. Assume that $x = Bx + d$. (a) Derive an equation for $e^{(k)} = y^{(k)} - x$. (b) Assume $y^{(1)} = By^{(0)} + d$. Show that $e^{(k)} = p_k(B)e^{(0)}$ where p_k is an even polynomial if k is even and an odd polynomial if k is odd. (c) Write $f^{(k)} = Q^Te^{(k)}$. Derive a difference equation for $f_j^{(k)}$ for $j = 1{:}n$. Try to specify the exact solution for general $f_j^{(0)}$ and $f_j^{(1)}$. (d) Show how to determine an optimal ω.

Notes and References for Sec. 10.1

As we mentioned, Young (1971) has the most comprehensive treatment of the SOR method. The object of "SOR theory" is to guide the user in choosing the relaxation parameter ω. In this setting, the ordering of equations and unknowns is critical. See

M.J.M. Bernal and J.H. Verner (1968). "On Generalizing of the Theory of Consistent Orderings for Successive Over-Relaxation Methods," *Numer. Math. 12,* 215–22.

D.M. Young (1970). "Convergence Properties of the Symmetric and Unsymmetric Over-Relaxation Methods," *Math. Comp. 24,* 793–807.

D.M. Young (1972). "Generalization of Property A and Consistent Ordering," *SIAM J. Num. Anal. 9,* 454–63.

R.A. Nicolaides (1974). "On a Geometrical Aspect of SOR and the Theory of Consistent Ordering for Positive Definite Matrices," *Numer. Math. 12,* 99–104.

L. Adams and H. Jordan (1986). "Is SOR Color-Blind?" *SIAM J. Sci. Stat. Comp. 7,* 490–506.

M. Eiermann and R.S. Varga (1993). "Is the Optimal ω Best for the SOR Iteration Method," *Lin. Alg. and Its Applic. 182,* 257–277.

An analysis of the Chebyshev semi-iterative method appears in

G.H. Golub and R.S. Varga (1961). "Chebychev Semi-Iterative Methods, Successive Over-Relaxation Iterative Methods, and Second-Order Richardson Iterative Methods, Parts I and II," *Numer. Math. 3,* 147–56, 157–68.

This work is premised on the assumption that the underlying iteration matrix has real eigenvalues. How to proceed when this is not the case is discussed in

T.A. Manteuffel (1977). "The Tchebychev Iteration for Nonsymmetric Linear Systems," *Numer. Math. 28*, 307–27.

M. Eiermann and W. Niethammer (1983). "On the Construction of Semi-iterative Methods," *SIAM J. Numer. Anal. 20*, 1153–1160.

W. Niethammer and R.S. Varga (1983). "The Analysis of k-step Iterative Methods for Linear Systems from Summability Theory," *Numer. Math. 41*, 177–206.

G.H. Golub and M. Overton (1988). "The Convergence of Inexact Chebychev and Richardson Iterative Methods for Solving Linear Systems," *Numer. Math. 53*, 571–594.

D. Calvetti, G.H. Golub, and L. Reichel (1994). "An Adaptive Chebyshev Iterative Method for Nonsymmetric Linear Systems Based on Modified Moments," *Numer. Math. 67*, 21 40.

Other unsymmetric methods include

M. Eiermann, W. Niethammer, and R.S. Varga (1992). "Acceleration of Relaxation Methods for Non-Hermitian Linear Systems," *SIAM J. Matrix Anal. Appl. 13*, 979–991.

H. Elman and G.H. Golub (1990). "Iterative Methods for Cyclically Reduced Non-Self-Adjoint Linear Systems I," *Math. Comp. 54*, 671–700.

H. Elman and G.H. Golub (1990). "Iterative Methods for Cyclically Reduced Non-Self-Adjoint Linear Systems II," *Math. Comp. 56*, 215–242.

R. Bramley and A. Sameh (1992). "Row Projection Methods for Large Nonsymmetric Linear Systems," *SIAM J. Sci. Statist. Comput. 13*, 168–193.

Sometimes it is possible to "symmetrize" an iterative method, thereby simplifying the acceleration process, since all the relevant eigenvalues are real. This is the case for the SSOR method discussed in

J.W. Sheldon (1955). "On the Numerical Solution of Elliptic Difference Equations," *Math. Tables Aids Comp. 9*, 101–12.

The parallel implementation of the classical iterations has received some attention. See

D.J. Evans (1984). "Parallel SOR Iterative Methods," *Parallel Computing 1*, 3 18.

N. Patel and H. Jordan (1984). "A Parallelized Point Rowwise Successive Over-Relaxation Method on a Multiprocessor," *Parallel Computing 1*, 207–222.

R.J. Plemmons (1986). "A Parallel Block Iterative Scheme Applied to Computations in Structural Analysis," *SIAM J. Alg. and Disc. Methods 7*, 337–347.

C. Kamath and A. Sameh (1989). "A Projection Method for Solving Nonsymmetric Linear Systems on Multiprocessors," *Parallel Computing 9*, 291–312.

We have seen that the condition $\kappa(A)$ is an important issue when direct methods are applied to $Ax = b$. However, the condition of the system also has a bearing on iterative method performance. See

M. Arioli and F. Romani (1985). "Relations Between Condition Numbers and the Convergence of the Jacobi Method for Real Positive Definite Matrices," *Numer. Math. 46*, 31 42.

M. Arioli, I.S. Duff, and D. Ruiz (1992). "Stopping Criteria for Iterative Solvers," *SIAM J. Matrix Anal. Appl. 13*, 138–144.

Iterative methods for singular systems are discussed in

A. Dax (1990). "The Convergence of Linear Stationary Iterative Processes for Solving Singular Unstructured Systems of Linear Equations," *SIAM Review 32*, 611–635.

Finally, the effect of rounding errors on the methods of this section are treated in

H. Wozniakowski (1978). "Roundoff-Error Analysis of Iterations for Large Linear Systems," *Numer. Math. 30*, 301–314.

P.A. Knight (1993). "Error Analysis of Stationary Iteration and Associated Problems," Ph.D. thesis, Department of Mathematics, University of Manchester, England.

10.2 The Conjugate Gradient Method

A difficulty associated with the SOR, Chebyshev semi-iterative, and related methods is that they depend upon parameters that are sometimes hard to choose properly. For example, the Chebyshev acceleration scheme needs good estimates of the largest and smallest eigenvalue of the underlying iteration matrix $M^{-1}N$. Unless this matrix is sufficiently structured, it may be analytically impossible and/or computationally expensive to do this.

In this section, we present a method without this difficulty for the symmetric positive definite $Ax = b$ problem, the well-known Hestenes-Stiefel conjugate gradient method. We derived this method in §9.3.1 from the Lanczos algorithm. The derivation now is from a different point of view and it will set the stage for various important generalizations in §10.3 and §10.4.

10.2.1 Steepest Descent

The starting point in the derivation is to consider how we might go about minimizing the function

$$\phi(x) = \frac{1}{2}x^TAx - x^Tb$$

where $b \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$ is assumed to be positive definite and symmetric. The minimum value of $\phi(x)$ is $-b^TA^{-1}b/2$, achieved by setting $x = A^{-1}b$. Thus, minimizing ϕ and solving $Ax = b$ are equivalent problems if A is symmetric positive definite.

One of the simplest strategies for minimizing ϕ is the *method of steepest descent.* At a current point x_c the function ϕ decreases most rapidly in the direction of the negative gradient: $-\nabla\phi(x_c) = b - Ax_c$. We call

$$r_c = b - Ax_c$$

the *residual* of x_c. If the residual is nonzero, then there exists a positive α such that $\phi(x_c + \alpha r_c) < \phi(x_c)$. In the method of steepest descent (with

exact line search) we set $\alpha = r_c^T r_c / r_c^T A r_c$ thereby minimizing

$$\phi(x_c + \alpha r_c) = \phi(x_c) - \alpha r_c^T r_c + \frac{1}{2}\alpha^2 r_c^T A r_c.$$

This gives

$$\begin{array}{l}
x_0 = \text{initial guess} \\
r_0 = b - Ax_0 \\
k = 0 \\
\textbf{while } r_k \neq 0 \\
\quad k = k + 1 \\
\quad \alpha_k = r_{k-1}^T r_{k-1} / r_{k-1}^T A r_{k-1} \\
\quad x_k = x_{k-1} + \alpha_k r_{k-1} \\
\quad r_k = b - Ax_k \\
\textbf{end}
\end{array} \qquad (10.2.1)$$

It can be shown that

$$\left(\phi(x_k) + \frac{1}{2}b^T A^{-1} b\right) \leq \left(1 - \frac{1}{\kappa_2(A)}\right)\left(\phi(x_{k-1}) + \frac{1}{2}b^T A^{-1} b\right) \qquad (10.2.2)$$

which implies global convergence. Unfortunately, the rate of convergence may be prohibitively slow if the condition $\kappa_2(A) = \lambda_1(A)/\lambda_n(A)$ is large. Geometrically this means that the level curves of ϕ are very elongated hyperellipsoids and minimization corresponds to finding the lowest point in a relatively flat, steep-sided valley. In steepest descent, we are forced to traverse back and forth *across* the valley rather than *down* the valley. Stated another way, the gradient directions that arise during the iteration are not different enough.

10.2.2 General Search Directions

To avoid the pitfalls of steepest descent, we consider the successive minimization of ϕ along a set of directions $\{p_1, p_2, \ldots\}$ that do not necessarily correspond to the residuals $\{r_0, r_1, \ldots\}$. It is easy to show that $\phi(x_{k-1} + \alpha p_k)$ is minimized by setting

$$\alpha = \alpha_k = p_k^T r_{k-1} / p_k^T A p_k.$$

With this choice it can be shown that

$$\phi(x_{k-1} + \alpha_k p_k) = \phi(x_{k-1}) - \frac{1}{2}\frac{(p_k^T r_{k-1})^2}{p_k^T A p_k}. \qquad (10.2.3)$$

To ensure a reduction in the size of ϕ we insist that p_k *not* be orthogonal to r_{k-1}. This leads to the following framework:

$$
\begin{aligned}
&x_0 = \text{initial guess} \\
&r_0 = b - Ax_0 \\
&k = 0 \\
&\textbf{while } r_k \neq 0 \\
&\qquad k = k + 1 \\
&\qquad \text{Choose a direction } p_k \text{ such that } p_k^T r_{k-1} \neq 0. \\
&\qquad \alpha_k = p_k^T r_{k-1} / p_k^T A p_k \\
&\qquad x_k = x_{k-1} + \alpha_k p_k \\
&\qquad r_k = b - Ax_k \\
&\textbf{end}
\end{aligned}
\tag{10.2.4}
$$

Note that

$$x_k \in x_0 + \text{span}\{p_1, \ldots, p_k\} \equiv \{x_0 + \gamma_1 p_1 + \cdots + \gamma_k p_k : \gamma_i \in \mathbb{R}\}.$$

Our goal is to choose the search directions in a way that guarantees convergence without the shortcomings of steepest descent.

10.2.3 A-Conjugate Search Directions

If the search directions are linearly independent *and* x_k solves the problem

$$\min_{x \in x_0 + \text{span}\{p_1, \ldots, p_k\}} \phi(x) \tag{10.2.5}$$

for $k = 1, 2, \ldots$, then convergence is guaranteed in at most n steps. This is because x_n minimizes ϕ over $\mathbb{R}^n$ and therefore satisfies $Ax_n = b$.

However, for this to be a viable approach the search directions must have the property that it is "easy" to compute x_k given x_{k-1}. Let us see what this says about the determination of p_k. If

$$x_k = x_0 + P_{k-1} y + \alpha p_k$$

where $P_{k-1} = [\, p_1, \ldots, p_{k-1} \,]$, $y \in \mathbb{R}^{k-1}$, and $\alpha \in \mathbb{R}$, then

$$\phi(x_k) = \phi(x_0 + P_{k-1} y) + \alpha y^T P_{k-1}^T A p_k + \frac{\alpha^2}{2} p_k^T A p_k - \alpha p_k^T r_0.$$

If $p_k \in \text{span}\{Ap_1, \ldots, Ap_{k-1}\}^{\perp}$, then the cross term $\alpha y^T P_{k-1}^T A p_k$ is zero and the search for the minimizing x_k splits into a pair of uncoupled minimizations, one for y and one for α:

$$
\begin{aligned}
\min_{x_k \in x_0 + \text{span}\{p_1, \ldots, p_k\}} \phi(x_k) &= \min_{y, \alpha} \phi(x_0 + P_{k-1} y + \alpha p_k) \\
&= \min_{y, \alpha} \left(\phi(x_0 + P_{k-1} y) + \frac{\alpha^2}{2} p_k^T A p_k - \alpha p_k^T r_0 \right)
\end{aligned}
$$

$$= \min_y \ \phi(x_0 + P_{k-1}y) + \min_\alpha \left(\frac{\alpha^2}{2} p_k^T A p_k - \alpha p_k^T r_0\right).$$

Note that if y_{k-1} solves the first min problem then $x_{k-1} = x_0 + P_{k-1}y_{k-1}$ minimizes ϕ over $x_0 + \text{span}\{p_1, \ldots, p_{k-1}\}$. The solution to the α min problem is given by $\alpha_k = p_k^T r_0 / p_k^T A p_k$. Note that because of A-conjugacy,

$$\begin{aligned} p_k^T r_{k-1} &= p_k^T (b - Ax_{k-1}) \\ &= p_k^T (b - A(x_0 + P_{k-1}y_{k-1})) = p_k^T r_0. \end{aligned}$$

With these results it follows that $x_k = x_{k-1} + \alpha_k p_k$ and we obtain the following instance of (10.2.4):

x_0 = initial guess
$k = 0$
$r_0 = b - Ax_0$
while $r_k \neq 0$
  $k = k + 1$
  Choose $p_k \in \text{span}\{Ap_1, \ldots, Ap_{k-1}\}^\perp$ so $p_k^T r_{k-1} \neq 0$. (10.2.6)
  $\alpha_k = p_k^T r_{k-1} / p_k^T A p_k$
  $x_k = x_{k-1} + \alpha_k p_k$
  $r_k = b - Ax_k$
end

The following lemma shows that it is possible to find the search directions with the required properties.

Lemma 10.2.1 *If $r_{k-1} \neq 0$, then there exists a $p_k \in$ span$\{Ap_1, \ldots, Ap_{k-1}\}^\perp$ such that $p_k^T r_{k-1} \neq 0$.*

Proof. For the case $k = 1$, set $p_1 = r_0$. If $k > 1$, then since $r_{k-1} \neq 0$ it follows that

$$\begin{aligned} A^{-1}b \notin x_0 + \text{span}\{p_1, \ldots, p_{k-1}\} \ &\Rightarrow\ b \notin Ax_0 + \text{span}\{Ap_1, \ldots, Ap_{k-1}\} \\ &\Rightarrow\ r_0 \notin \text{span}\{Ap_1, \ldots, Ap_{k-1}\}. \end{aligned}$$

Thus there exists a $p \in \text{span}\{Ap_1, \ldots, Ap_{k-1}\}^\perp$ such that $p^T r_0 \neq 0$. But $x_{k-1} \in x_0 + \text{span}\{p_1, \ldots, p_{k-1}\}$ and so $r_{k-1} \in r_0 + \text{span}\{Ap_1, \ldots, Ap_{k-1}\}$. It follows that $p^T r_{k-1} = p^T r_0 \neq 0$. □

The search directions in (10.2.6) are said to be *A-conjugate* because $p_i^T A p_j = 0$ for all $i \neq j$. Note that if $P_k = [p_1, \ldots, p_k]$ is the matrix of these vectors, then

$$P_k^T A P_k = \text{diag}(p_1^T A p_1, \ldots, p_k^T A p_k)$$

is nonsingular since A is positive definite and the search directions are nonzero. It follows that P_k has full column rank. This guarantees convergence in (10.2.6) in at most n steps because x_n (if we get that far) minimizes $\phi(x)$ over $\text{ran}(P_n) = \mathbb{R}^n$.

10.2.4 Choosing a Best Search Direction

A way to combine the positive aspects of steepest descent and A-conjugate searching is to choose p_k in (10.2.6) to be the closest vector to r_{k-1} that is A-conjugate to $p_1, \ldots, p_{k-1}$. This defines "version zero" of the *method of conjugate gradients*:

x_0 = initial guess
$k = 0$
$r_0 = b - Ax_0$
while $r_k \neq 0$
  $k = k + 1$
  if $k = 1$
    $p_1 = r_0$
  else (10.2.7)
    Let p_k minimize $\| p - r_{k-1} \|_2$ over all vectors
      $p \in \text{span}\{Ap_1, \ldots, Ap_{k-1}\}^{\perp}$
  end
  $\alpha_k = p_k^T r_{k-1} / p_k^T A p_k$
  $x_k = x_{k-1} + \alpha_k p_k$
  $r_k = b - Ax_k$
end
$x = x_k$

To make this an effective sparse $Ax = b$ solver, we need an efficient method for computing p_k. A considerable amount of analysis is required to develop the final recursions. The first step is to show that p_k is the minimum residual of a certain least squares problem.

Lemma 10.2.2 *For $k \geq 2$ the vectors p_k generated by (10.2.7) satisfy*

$$p_k = r_{k-1} - AP_{k-1}z_{k-1},$$

where $P_{k-1} = [p_1, \ldots, p_{k-1}]$ and z_{k-1} solves $\displaystyle\min_{z \in \mathbb{R}^{k-1}} \| r_{k-1} - AP_{k-1}z \|_2$.

Proof. Suppose z_{k-1} solves the above LS problem and let p be the associated minimum residual:

$$p = r_{k-1} - AP_{k-1}z_{k-1}.$$

It follows that $p^T AP_{k-1} = 0$. Moreover, $p = [I - (AP_{k-1})(AP_{k-1})^+]r_{k-1}$ is the orthogonal projection of r_{k-1} into $\text{ran}(AP_{k-1})^{\perp}$ and so it is the closest vector in $\text{ran}(AP_{k-1})^{\perp}$ to r_{k-1}. Thus, $p = p_k$. □

With this result we can establish a number of important relationships between the residuals r_k, the search directions p_k, and the Krylov subspaces

$$\mathcal{K}(r_0, A, k) = \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}.$$

Theorem 10.2.3 *After k iterations in (10.2.7) we have*

$$\begin{aligned} r_k &= r_{k-1} - \alpha_k A p_k & (10.2.8)\\ P_k^T r_k &= 0 & (10.2.9)\\ \text{span}\{p_1, \ldots, p_k\} &= \text{span}\{r_0, \ldots, r_{k-1}\} = \mathcal{K}(r_0, A, k) & (10.2.10)\end{aligned}$$

and the residuals $r_0, \ldots, r_k$ are mutually orthogonal.

Proof. Equation (10.2.8) follows by applying A to both sides of $x_k = x_{k-1} + \alpha_k p_k$ and using the definition of the residual.

To prove (10.2.9), we recall that $x_k = x_0 + P_k y_k$ where y_k is the minimizer of

$$\phi(x_0 + P_k y) = \phi(x_0) + \frac{1}{2} y^T (P_k^T A P_k) y - y^T P_k (b - Ax_0).$$

But this means that y_k solves the linear system $(P_k^T A P_k)y = P_k^T(b - Ax_0)$. Thus

$$0 = P_k^T(b - Ax_0) - P_k^T A P_k y_k = P_k^T(b - A(x_0 + P_k y_k)) = P_k^T r_k.$$

To prove (10.2.10) we note from (10.2.8) that

$$\{Ap_1, \ldots, Ap_{k-1}\} \subseteq \text{span}\{r_0, \ldots, r_{k-1}\}$$

and so from Lemma 10.2.2,

$$p_k = r_{k-1} - [Ap_1, \ldots, Ap_{k-1}]\, z_{k-1} \in \text{span}\{r_0, \ldots, r_{k-1}\}$$

It follows that

$$[p_1, \ldots, p_k] = [r_0, \ldots, r_{k-1}]\, T$$

for some upper triangular T. Since the search directions are independent, T is nonsingular. This shows

$$\text{span}\{p_1, \ldots, p_k\} = \text{span}\{r_0, \ldots, r_{k-1}\}.$$

Using (10.2.8) we see that

$$r_k \in \text{span}\{r_{k-1}, Ap_k\} \subseteq \text{span}\{r_{k-1}, Ar_0, \ldots, Ar_{k-1}\}.$$

The Krylov space connection in (10.2.10)follows from this by induction.

Finally, to establish the mutual orthogonality of the residuals, we note from (10.2.9) that r_k is orthogonal to any vector in the range of P_k. But from (10.2.10) this subspace contains $r_0, \ldots, r_{k-1}$. □

Using these facts we next show that p_k is a simple linear combination of its predecessor p_{k-1} and the "current" residual r_{k-1}.

Corollary 10.2.4 *The residuals and search directions in (10.2.7) have the property that* $p_k \in \text{span}\{p_{k-1}, r_{k-1}\}$ *for* $k \geq 2$.

Proof. If $k = 2$, then from (10.2.10) $p_2 \in \text{span}\{r_0, r_1\}$. But $p_1 = r_0$ and so p_2 is a linear combination of p_1 and r_1.

If $k > 2$, then partition the vector z_{k-1} of Lemma 10.2.2 as

$$z_{k-1} = \begin{bmatrix} w \\ \mu \end{bmatrix} \begin{matrix} k-2 \\ 1 \end{matrix} .$$

Using the identity $r_{k-1} = r_{k-2} - \alpha_{k-1}Ap_{k-1}$, we see that

$$\begin{aligned} p_k &= r_{k-1} - AP_{k-1}z_{k-1} = r_{k-1} - AP_{k-2}w - \mu Ap_{k-1} \\ &= \left(1 + \frac{\mu}{\alpha_{k-1}}\right) r_{k-1} + s_{k-1} \end{aligned}$$

where

$$\begin{aligned} s_{k-1} &= -\frac{\mu}{\alpha_{k-1}} r_{k-2} - AP_{k-2}w \\ &\in \text{span}\{r_{k-2}, AP_{k-2}w\} \\ &\subseteq \text{span}\{r_{k-2}, Ap_1, \ldots, Ap_{k-2}\} \\ &\subseteq \text{span}\{r_1, \ldots, r_{k-2}\} \end{aligned}$$

Because the r_i are mutually orthogonal, it follows that s_{k-1} and r_{k-1} are orthogonal to each other. Thus, the least squares problem of Lemma 10.2.2 boils down to choosing μ and w such that

$$\| p_k \|_2^2 = \left(1 + \frac{\mu}{\alpha_{k-1}}\right)^2 \| r_{k-1} \|_2^2 + \| s_{k-1} \|_2^2$$

is minimum. Since the 2-norm of $r_{k-2} - AP_{k-2}z$ is minimized by z_{k-2} giving residual p_{k-1}, it follows that s_{k-1} is a multiple of p_{k-1}. Consequently, $p_k \in \text{span}\{r_{k-1}, p_{k-1}\}$. □

We are now set to derive a very simple expression for p_k. Without loss of generality we may assume from Corollary 10.2.4 that

$$p_k = r_{k-1} + \beta_k p_{k-1}.$$

Since $p_{k-1}^T Ap_k = 0$ it follows that

$$\beta_k = -\frac{p_{k-1}^T Ar_{k-1}}{p_{k-1}^T Ap_{k-1}}$$

This leads us to "version 1" of the conjugate gradient method:

$$
\begin{array}{l}
x_0 = \text{initial guess} \\
k = 0 \\
r_0 = b - Ax_0 \\
\textbf{while } r_k \neq 0 \\
\quad k = k + 1 \\
\quad \textbf{if } k = 1 \\
\quad\quad p_1 = r_0 \\
\quad \textbf{else} \\
\quad\quad \beta_k = -p_{k-1}^T A r_{k-1} / p_{k-1}^T A p_{k-1} \\
\quad\quad p_k = r_{k-1} + \beta_k p_{k-1} \\
\quad \textbf{end} \\
\quad \alpha_k = p_k^T r_{k-1} / p_k^T A p_k \\
\quad x_k = x_{k-1} + \alpha_k p_k \\
\quad r_k = b - Ax_k \\
\textbf{end} \\
x = x_k
\end{array}
\qquad (10.2.11)
$$

In this implementation, the method requires three separate matrix-vector multiplications per step. However, by computing residuals recursively via $r_k = r_{k-1} - \alpha_k A p_k$ and substituting

$$r_{k-1}^T r_{k-1} = -\alpha_{k-1} r_{k-1}^T A p_{k-1} \qquad (10.2.12)$$

and

$$r_{k-2}^T r_{k-2} = \alpha_{k-1} p_{k-1}^T A p_{k-1} \qquad (10.2.13)$$

into the formula for β_k, we obtain the following more efficient version:

Algorithm 10.2.1 [Conjugate Gradients] If $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite, $b \in \mathbb{R}^n$, and $x_0 \in \mathbb{R}^n$ is an initial guess ($Ax_0 \approx b$), then the following algorithm computes $x \in \mathbb{R}^n$ so $Ax = b$.

$$
\begin{array}{l}
k = 0 \\
r_0 = b - Ax_0 \\
\textbf{while } r_k \neq 0 \\
\quad k = k + 1 \\
\quad \textbf{if } k = 1 \\
\quad\quad p_1 = r_0 \\
\quad \textbf{else} \\
\quad\quad \beta_k = r_{k-1}^T r_{k-1} / r_{k-2}^T r_{k-2} \\
\quad\quad p_k = r_{k-1} + \beta_k p_{k-1} \\
\quad \textbf{end} \\
\quad \alpha_k = r_{k-1}^T r_{k-1} / p_k^T A p_k \\
\quad x_k = x_{k-1} + \alpha_k p_k \\
\quad r_k = r_{k-1} - \alpha_k A p_k \\
\textbf{end} \\
x = x_k
\end{array}
$$

This procedure is essentially the form of the conjugate gradient algorithm that appears in the original paper by Hestenes and Stiefel (1952). Note that only one matrix-vector multiplication is required per iteration.

10.2.5 The Lanczos Connection

In §9.3.1 we derived the conjugate gradient method from the Lanczos algorithm. Now let us look at the connections between these two algorithms in the reverse direction by "deriving" the Lanczos process from conjugate gradients. Define the matrix of residuals $R_k \in \mathbb{R}^{n\times k}$ by

$$R_k = [\, r_0, \ldots, r_{k-1} \,]$$

and the upper bidiagonal matrix $B_k \in \mathbb{R}^{k\times k}$ by

$$B_k = \begin{bmatrix} 1 & -\beta_2 & 0 & \cdots & 0 \\ 0 & 1 & -\beta_3 & & \vdots \\ & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\beta_k \\ 0 & \cdots & & 0 & 1 \end{bmatrix}.$$

From the equations $p_i = r_{i-1} + \beta_i p_{i-1}$, $i = 2{:}k$, and $p_1 = r_0$ it follows that $R_k = P_k B_k$. Since the columns of $P_k = [\, p_1, \ldots, p_k \,]$ are A-conjugate, we see that $R_k^T A R_k = B_k^T \mathrm{diag}(p_1^T A p_1, \ldots, p_k^T A p_k) B_k$ is tridiagonal. From (10.2.10) it follows that if

$$\Delta = \mathrm{diag}(\rho_0, \ldots, \rho_{k-1}) \qquad \rho_i = \| \, r_i \, \|_2$$

then the columns of $R_k \Delta^{-1}$ form an orthonormal basis for the subspace $\mathrm{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$. Consequently, the columns of this matrix are essentially the Lanczos vectors of Algorithm 9.3.1, i.e.,

$$q_i = \pm r_{i-1}/\rho_{i-1} \qquad i = 1{:}k.$$

Moreover, the tridiagonal matrix associated with these Lanczos vectors is given by

$$T_k = \Delta^{-1} B_k^T \mathrm{diag}(p_i^T A p_i) B_k \Delta^{-1}. \qquad (10.2.14)$$

The diagonal and subdiagonal of this matrix involve quantities that are readily available during the conjugate gradient iteration. Thus, we can obtain good estimates of A's extremal eigenvalues (and condition number) as we generate the x_k in Algorithm 10.2.1.

10.2.6 Some Practical Details

The termination criteria in Algorithm 10.2.1 is unrealistic. Rounding errors lead to a loss of orthogonality among the residuals and finite termination is not mathematically guaranteed. Moreover, when the conjugate gradient method is applied, n is usually so big that $O(n)$ iterations represents an unacceptable amount of work. As a consequence of these observations, it is customary to regard the method as a genuinely iterative technique with termination based upon an iteration maximum k_{max} and the residual norm. This leads to the following practical version of Algorithm 10.2.1:

x = initial guess
$k = 0$
$r = b - Ax_0$
$\rho_0 = \| r \|_2^2$
while $(\sqrt{\rho_k} > \epsilon \| b \|_2) \wedge (k < k_{max})$
    $k = k + 1$
    if $k = 1$
        $p = r$
    else (10.2.16)
        $\beta_k = \rho_{k-1}/\rho_{k-2}$
        $p = r + \beta_k p$
    end
    $w = Ap$
    $\alpha_k = \rho_{k-1}/p^T w$
    $x = x + \alpha_k p$
    $r = r - \alpha_k w$
    $\rho_k = \| r \|_2^2$
end

This algorithm requires one matrix-vector multiplication and 10n flops per iteration. Notice that just four n-vectors of storage are essential: x, r, p, and w. The subscripting of the scalars is not necessary and is only done here to facilitate comparison with Algorithm 10.2.1.

It is also possible to base the termination criteria on heuristic estimates of the error $A^{-1}r_k$ by approximating $\| A^{-1} \|_2$ with the reciprocal of the smallest eigenvalue of the tridiagonal matrix T_k given in (10.2.14).

The idea of regarding conjugate gradients as an iterative method began with Reid (1971). The iterative point of view is useful but then the *rate* of convergence is central to the method's success.

10.2.7 Convergence Properties

We conclude this section by examining the convergence of the conjugate gradient iterates $\{x_k\}$. Two results are given and they both say that the

method performs well when A is near the identity either in the sense of a low rank perturbation or in the sense of norm.

Theorem 10.2.5 *If $A = I + B$ is an n-by-n symmetric positive definite matrix and rank$(B) = r$ then Algorithm 10.2.1 converges in at most $r+1$ steps.*

Proof. The dimension of

$$\text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\} = \text{span}\{r_0, Br_0, \ldots, B^{k-1}r_0\}$$

cannot exceed $r+1$. Since $p_1, \ldots, p_k$ span this subspace and are independent, the iteration cannot progress beyond $r+1$ steps. □

An important metatheorem follows from this result:

- If A is close to a rank r correction to the identity, then Algorithm 10.2.1 almost converges after $r+1$ steps.

We show how this heuristic can be exploited in the next section.

An error bound of a different flavor can be obtained in terms of the A-norm which we define as follows:

$$\| w \|_A = \sqrt{w^T A w}\,.$$

Theorem 10.2.6 *Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $b \in \mathbb{R}^n$ If Algorithm 10.2.1 produces iterates $\{x_k\}$ and $\kappa = \kappa_2(A)$ then*

$$\| x - x_k \|_A \leq 2\| x - x_0 \|_A \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k .$$

Proof. See Luenberger (1973, p.187). □

The accuracy of the $\{x_k\}$ is often much better than this theorem predicts. However, a heuristic version of Theorem 10.2.6 turns out to be very useful:

- The conjugate gradient method converges very fast in the A-norm if $\kappa_2(A) \approx 1$.

In the next section we show how we can sometimes convert a given $Ax = b$ problem into a related $\tilde{A}\tilde{x} = \tilde{b}$ problem with $\tilde{A}$ being close to the identity.

Problems

P10.2.1 Verify that the residuals in (10.2.1) satisfy $r_i^T r_j = 0$ whenever $j = i + 1$.

P10.2.2 Verify (10.2.2).

P10.2.3 Verify (10.2.3).

P10.2.4 Verify (10.2.12) and (10.2.13).

P10.2.5 Give formula for the entries of the tridiagonal matrix T_k in (10.2.14).

P10.2.6 Compare the work and storage requirements associated with the practical implementation of Algorithms 9.3.1 and 10.2.1.

P10.2.7 Show that if $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite and has k distinct eigenvalues, then the conjugate gradient method does not require more than $k+1$ steps to converge.

P10.2.8 Use Theorem 10.2.6 to verify that

$$\| x_k - A^{-1}b \|_2 \le 2\sqrt{\kappa}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \| x_0 - A^{-1}b \|_2.$$

Notes and References for Sec. 10.2

The conjugate gradient method is a member of a larger class of methods that are referred to as *conjugate direction* algorithms. In a conjugate direction algorithm the search directions are all B-conjugate for some suitably chosen matrix B. A discussion of these methods appears in

J.E. Dennis Jr. and K. Turner (1987). "Generalized Conjugate Directions," *Lin. Alg. and Its Applic. 88/89*, 187–209.

G.W. Stewart (1973). "Conjugate Direction Methods for Solving Systems of Linear Equations," *Numer. Math. 21*, 284–97.

Some historical and unifying perspectives are offered in

G. Golub and D. O'Leary (1989). "Some History of the Conjugate Gradient and Lanczos Methods," *SIAM Review 31*, 50–102.

M.R. Hestenes (1990). "Conjugacy and Gradients," in *A History of Scientific Computing*, Addison-Wesley, Reading, MA.

S. Ashby, T.A. Manteuffel, and P.E. Saylor (1992). "A Taxonomy for Conjugate Gradient Methods," *SIAM J. Numer. Anal. 27*, 1542–1568.

The classic reference for the conjugate gradient method is

M.R. Hestenes and E. Stiefel (1952). "Methods of Conjugate Gradients for Solving Linear Systems," *J. Res. Nat. Bur. Stand. 49*, 409–36.

An exact arithmetic analysis of the method may be found in chapter 2 of

M.R. Hestenes (1980). *Conjugate Direction Methods in Optimization*, Springer-Verlag, Berlin.

See also

O. Axelsson (1977). "Solution of Linear Systems of Equations: Iterative Methods," in *Sparse Matrix Techniques: Copenhagen*, 1976, ed. V.A. Barker, Springer-Verlag, Berlin.

For a discussion of conjugate gradient convergence behavior, see

D. G. Luenberger (1973). *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, New York.

A. van der Sluis and H.A. Van Der Vorst (1986). "The Rate of Convergence of Conjugate Gradients," *Numer. Math. 48*, 543–560.

The idea of using the conjugate gradient method as an iterative method was first discussed in

J.K. Reid (1971). " On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations," *in Large Sparse Sets of Linear Equations* , ed. J.K. Reid, Academic Press, New York, pp. 231–54.

Several authors have attempted to explain the algorithm's behavior in finite precision arithmetic. See

H. Wozniakowski (1980). "Roundoff Error Analysis of a New Class of Conjugate Gradient Algorithms," *Lin. Alg. and Its Applic. 29,*

A. Greenbaum and Z. Strakos (1992). "Predicting the Behavior of Finite Precision lanczos and Conjugate Gradient Computations," *SIAM J. Matrix Ana. Applic. 13,* 121–137.

See also the analysis in

G.W. Stewart (1975). "The Convergence of the Method of Conjugate Gradients at Isolated Extreme Points in the Spectrum," *Numer. Math. 24,* 85–93.

A. Jennings (1977). "Influence of the Eigenvalue Spectrum on the Convergence Rate of the Conjugate Gradient Method," *J. Inst. Math. Applic. 20,* 61–72.

J. Cullum and R. Willoughby (1980). "The Lanczos Phenomena: An Interpretation Based on Conjugate Gradient Optimization," *Lin. Alg. and Its Applic. 29,* 63–90.

Finally, we mention that the method can be used to compute an eigenvector of a large sparse symmetric matrix:

A. Ruhe and T. Wiberg (1972). "The Method of Conjugate Gradients Used in Inverse Iteration," *BIT 12,* 543–54.

10.3 Preconditioned Conjugate Gradients

We concluded the previous section by observing that the method of conjugate gradients works well on matrices that are either well conditioned or have just a few distinct eigenvalues. (The latter being the case when A is a lower rank perturbation of the identity.) In this section we show how to *precondition* a linear system so that the matrix of coefficients assumes one of these nice forms. Our treatment is quite brief and informal. Golub and Meurant (1983) and Axelsson (1985) have more comprehensive expositions.

10.3.1 Derivation

Consider the n-by-n symmetric positive definite linear system $Ax = b$. The idea behind preconditioned conjugate gradients is to apply the "regular" conjugate gradient method to the transformed system

$$\tilde{A}\tilde{x} = \tilde{b}, \tag{10.3.1}$$

where $\tilde{A} = C^{-1}AC^{-1}$, $\tilde{x} = Cx$, $\tilde{b} = C^{-1}b$, and C is symmetric positive definite. In view of our remarks in §10.2.8, we should try to choose C

so that $\tilde{A}$ is well conditioned or a matrix with clustered eigenvalues. For reasons that will soon emerge, the matrix C^2 must also be "simple."

If we apply Algorithm 10.2.1 to (10.3.1), then we obtain the iteration

$$
\begin{array}{l}
k = 0 \\
\tilde{x}_0 = \text{initial guess } (\tilde{A}\tilde{x}_0 \approx \tilde{b}) \\
\tilde{r}_0 = \tilde{b} - \tilde{A}\tilde{x}_0 \\
\textbf{while } \tilde{r}_k \neq 0 \\
\quad k = k + 1 \\
\quad \textbf{if } k = 1 \\
\quad\quad \tilde{p}_1 = \tilde{r}_0 \\
\quad \textbf{else} \\
\quad\quad \beta_k = \tilde{r}_{k-1}^T\tilde{r}_{k-1}/\tilde{r}_{k-2}^T\tilde{r}_{k-2} \\
\quad\quad \tilde{p}_k = \tilde{r}_{k-1} + \beta_k\tilde{p}_{k-1} \\
\quad \textbf{end} \\
\quad \alpha_k = \tilde{r}_{k-1}^T\tilde{r}_{k-1}/\tilde{p}_k^T C^{-1}AC^{-1}\tilde{p}_k \\
\quad \tilde{x}_k = \tilde{x}_{k-1} + \alpha_k\tilde{p}_k \\
\quad \tilde{r}_k = \tilde{r}_{k-1} - \alpha_k C^{-1}AC^{-1}\tilde{p}_k \\
\textbf{end} \\
\tilde{x} = \tilde{x}_k
\end{array}
\qquad (10.3.2)
$$

Here, $\tilde{x}_k$ should be regarded as an approximation to $\tilde{x}$ and $\tilde{r}_k$ is the residual in the transformed coordinates, i.e., $\tilde{r}_k = \tilde{b} - \tilde{A}\tilde{x}_k$. Of course, once we have $\tilde{x}$ then we can obtain x via the equation $x = C^{-1}\tilde{x}$. However, it is possible to avoid explicit reference to the matrix C^{-1} by defining $\tilde{p}_k = Cp_k$, $\tilde{x}_k = Cx_k$, and $\tilde{r}_k = C^{-1}r_k$. Indeed, if we substitute these definitions into (10.3.2) and recall that $\tilde{b} = C^{-1}b$ and $\tilde{x} = Cx$, then we obtain

$$
\begin{array}{l}
k = 0 \\
x_0 = \text{initial guess } (Ax_0 \approx b) \\
r_0 = b - Ax_0 \\
\textbf{while } C^{-1}r_k \neq 0 \\
\quad k = k + 1 \\
\quad \textbf{if } k = 1 \\
\quad\quad Cp_1 = C^{-1}r_0 \\
\quad \textbf{else} \\
\quad\quad \beta_k = (C^{-1}r_{k-1})^T(C^{-1}r_{k-1})/(C^{-1}r_{k-2})^T(C^{-1}r_{k-2}) \\
\quad\quad Cp_k = C^{-1}r_{k-1} + \beta_k Cp_{k-1} \\
\quad \textbf{end} \\
\quad \alpha_k = (C^{-1}r_{k-1})^T(C^{-1}r_{k-1})/(Cp_k)^T(C^{-1}AC^{-1})(Cp_k) \\
\quad Cx_k = Cx_{k-1} + \alpha_k Cp_k \\
\quad C^{-1}r_k = C^{-1}r_{k-1} - \alpha_k(C^{-1}AC^{-1})Cp_k \\
\textbf{end} \\
Cx = Cx_k
\end{array}
\qquad (10.3.3)
$$

If we define the *preconditioner* M by $M = C^2$ (also positive definite) and let z_k be the solution of the system $Mz_k = r_k$ then (10.3.3) simplifies to

Algorithm 10.3.1 [Preconditioned Conjugate Gradients] Given a symmetric positive definite $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, a symmetric positive definite preconditioner M, and an initial guess x_0 ($Ax_0 \approx b$), the following algorithm solves the linear system $Ax = b$.

> $k = 0$
> $r_0 = b - Ax_0$
> **while** $(r_k \neq 0)$
>   Solve $Mz_k = r_k$.
> $k = k + 1$
>   **if** $k = 1$
> $p_1 = z_0$
>   **else**
> $\beta_k = r_{k-1}^T z_{k-1} / r_{k-2}^T z_{k-2}$
> $p_k = z_{k-1} + \beta_k p_{k-1}$
>   **end**
> $\alpha_k = r_{k-1}^T z_{k-1} / p_k^T A p_k$
> $x_k = x_{k-1} + \alpha_k p_k$
> $r_k = r_{k-1} - \alpha_k A p_k$
> **end**
> $x = x_k$

A number of important observations should be made about this procedure:

- It can be shown that the residuals and search directions satisfy

$$r_j^T M^{-1} r_i = 0 \qquad i \neq j \tag{10.3.4}$$

$$p_j^T (C^{-1} A C^{-1}) p_i = 0 \qquad i \neq j \tag{10.3.5}$$

- The denominators $r_{k-2}^T z_{k-2} = z_{k-2}^T M z_{k-2}$ never vanish because M is positive definite.

- Although the transformation C figured heavily in the derivation of the algorithm, its action is only felt through the preconditioner $M = C^2$.

- For Algorithm 10.3.1 to be an effective sparse matrix technique, linear systems of the form $Mz = r$ must be easily solved *and* convergence must be rapid.

The choice of a good preconditioner can have a dramatic effect upon the rate of convergence. Some of the possibilities are now discussed.

10.3.2 Incomplete Cholesky Preconditioners

One of the most important preconditioning strategies involves computing an *incomplete Cholesky factorization* of A. The idea behind this approach is to calculate a lower triangular matrix H with the property that H has some tractable sparsity structure and is somehow "close" to A's exact Cholesky factor G. The preconditioner is then taken to be $M = HH^T$. To appreciate this choice consider the following facts:

- There exists a unique symmetric positive definite matrix C such that $M = C^2$.
- There exists an orthogonal Q such that $C = QH^T$, i.e., H^T is the upper triangular factor of a QR factorization of C.

We therefore obtain the heuristic

$$\begin{aligned} \tilde{A} &= C^{-1}AC^{-1} = C^{-T}AC^{-1} \\ &= (HQ^T)^{-1}A(QH^T)^{-1} = Q(H^{-1}GG^TH^{-T})Q^T \approx I \end{aligned} \tag{10.3.6}$$

Thus, the better H approximates G the smaller the condition of $\tilde{A}$, and the better the performance of Algorithm 10.3.1.

An easy but effective way to determine such a simple H that approximates G is to step through the Cholesky reduction setting h_{ij} to zero if the corresponding a_{ij} is zero. Pursuing this with the outer product version of Cholesky we obtain

```
for k = 1:n
    A(k,k) = sqrt(A(k,k))
    for i = k+1:n
        if A(i,k) ≠ 0
            A(i,k) = A(i,k)/A(k,k)
        end
    end                                              (10.3.7)
    for j = k+1:n
        for i = j:n
            if A(i,j) ≠ 0
                A(i,j) = A(i,j) - A(i,k)A(j,k)
            end
        end
    end
end
```

In practice, the matrix A and its incomplete Cholesky factor H would be stored in an appropriate data structure and the looping in the above algorithm would take on a very special appearance.

Unfortunately, (10.3.7) is not always stable. Classes of positive definite matrices for which incomplete Cholesky is stable are identified in Manteuffel (1979). See also Elman (1986).

10.3.3 Incomplete Block Preconditioners

As with just about everything else in this book, the incomplete factorization ideas outlined in the previous subsection have a block analog. We illustrate this by looking at the *incomplete block Cholesky factorization* of the symmetric, positive definite, block tridiagonal matrix

$$A = \begin{bmatrix} A_1 & E_1^T & 0 \\ E_1 & A_2 & E_2^T \\ 0 & E_2 & A_3 \end{bmatrix}.$$

For purposes of illustration, we assume that the A_i are tridiagonal and the E_i are diagonal. Matrices with this structure arise from the standard 5-point discretization of self-adjoint elliptic partial differential equations over a two-dimensional domain.

The 3-by-3 case is sufficiently general. Our discussion is based upon Concus, Golub, and Meurant (1985). Let

$$G = \begin{bmatrix} G_1 & 0 & 0 \\ F_1 & G_2 & 0 \\ 0 & F_2 & G_3 \end{bmatrix}$$

be the exact block Cholesky factor of A. Although G is sparse as a block matrix, the individual blocks are dense with the exception of G_1. This can be seen from the required computations:

$$\begin{aligned} G_1G_1^T &= B_1 \equiv A_1 \\ F_1 &= E_1G_1^{-1} \\ G_2G_2^T &= B_2 \equiv A_2 - F_1F_1^T = A_2 - E_1B_1^{-1}E_1^T \\ F_2 &= E_2G_2^{-1} \\ G_3G_3^T &= B_3 \equiv A_3 - F_2F_2^T = A_3 - E_2B_2^{-1}E_2^T \end{aligned}$$

We therefore seek an approximate block Cholesky factor of the form

$$\tilde{G} = \begin{bmatrix} \tilde{G}_1 & 0 & 0 \\ \tilde{F}_1 & \tilde{G}_2 & 0 \\ 0 & \tilde{F}_2 & \tilde{G}_3 \end{bmatrix}$$

so that we can easily solve systems that involve the preconditioner $M = \tilde{G}\tilde{G}^T$. This involves the imposition of sparsity on $\tilde{G}$'s blocks and here is a reasonable approach given that the A_i are tridiagonal and the E_i are diagonal:

$$\begin{aligned}
\tilde{G}_1\tilde{G}_1^T &= \tilde{B}_1 \equiv A_1 \\
\tilde{F}_1 &= E_1\tilde{G}_1^{-1} \\
\tilde{G}_2\tilde{G}_2^T &= \tilde{B}_2 = A_2 - E_1\Lambda_1E_1^T, \quad \Lambda_1 \text{ (tridiagonal) } \approx \tilde{B}_1^{-1} \\
\tilde{F}_2 &= E_2\tilde{G}_2^{-1} \\
\tilde{G}_3\tilde{G}_3^T &= \tilde{B}_3 \equiv A_3 - E_2\Lambda_2E_2^T, \quad \Lambda_2 \text{ (tridiagonal) } \approx \tilde{B}_2^{-1}
\end{aligned}$$

Note that all the $\tilde{B}_i$ are tridiagonal. Clearly, the Λ_i must be carefully chosen to ensure that the $\tilde{B}_i$ are also symmetric and positive definite. It then follows that the $\tilde{G}_i$ are lower bidiagonal. The $\tilde{F}_i$ are full, but they need not be explicitly formed. For example, in the course of solving the system $Mz = r$ we must solve a system of the form

$$\begin{bmatrix} \tilde{G}_1 & 0 & 0 \\ \tilde{F}_1 & \tilde{G}_2 & 0 \\ 0 & \tilde{F}_2 & \tilde{G}_3 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}.$$

Forward elimination can be used to carry out matrix-vector products that involve the $\tilde{F}_i = E_i\tilde{G}_i^{-1}$:

$$\begin{aligned}
\tilde{G}_1w_1 &= r_1 \\
\tilde{G}_2w_2 &= r_2 - \tilde{F}_1w_1 = r_2 - E_1\tilde{G}_1^{-1}w_1 \\
\tilde{G}_3w_3 &= r_3 - \tilde{F}_2w_2 = r_3 - E_2\tilde{G}_2^{-1}w_2
\end{aligned}$$

The choice of Λ_i is delicate as the resulting $\tilde{B}_i$ must be positive definite. As we have organized the computation, the central issue is how to approximate the inverse of an m-by-m symmetric, positive definite, tridiagonal matrix $T = (t_{ij})$ with a symmetric tridiagonal matrix Λ. There are several reasonable approaches:

- Set $\Lambda = \text{diag}(1/t_{11}, \ldots, 1/t_{nn})$.
- Take Λ to be the tridiagonal part of T^{-1}. This can be efficiently computed since there exist $u, v \in \mathbb{R}^m$ such that the lower triangular part of T^{-1} is the lower triangular part of uv^T. See Asplund(1959).
- Set $\Lambda = U^TU$ where U is the lower bidiagonal portion of G^{-1} where $T = GG^T$ is the Cholesky factorization. This can be found in $O(m)$ flops.

For a discussion of these approximations and what they imply about the associated preconditioners, see Concus, Golub, and Meurant (1985).

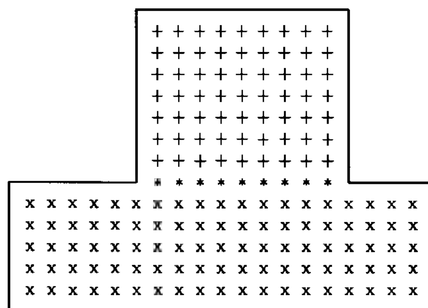
10.3.4 Domain Decomposition Ideas

The numerical solution of elliptic partial differential equations often leads to linear systems of the form

$$\begin{bmatrix} A_1 & \cdots & & \cdots & B_1 \\ \vdots & A_2 & & & B_2 \\ & & \ddots & & \vdots \\ \vdots & & & A_p & B_p \\ B_1^T & B_2^T & \cdots & B_p^T & Q \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \\ z \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \\ f \end{bmatrix} \tag{10.3.8}$$

if the unknowns are properly sequenced. See Meurant (1984). Here, the A_i are symmetric positive definite, the B_i are sparse, and the last block column is generally much narrower than the others.

An example with $p = 2$ serves to connect (10.3.8) and its block structure with the underlying problem geometry and the chosen *domain decomposition*. Suppose we are to solve Poisson's equation on the following domain:



With the usual discretization, an unknown at a mesh point is coupled only to its "north", "east", "south", and "west" neighbor. There are three "types" of variables: those interior to the top subdomain (aggregated in the subvector x_1 and associated with the "+" mesh points), those interior to the bottom subdomain (aggregated in the subvector x_2 and associated with the "x" mesh points), and those on the interface between the two subdomains (aggregated in the subvector z and associated with the "*" mesh points). Note that the interior unknowns of one subdomain are not coupled to the interior unknowns of another subdomain, which accounts

for the zero blocks in (10.3.8). Also observe that the number of interface unknowns is typically small compared to the overall number of unknowns.

Now let us explore the preconditioning possibilities associated with (10.3.8). We continue with the $p = 2$ case for simplicity. If we set

$$M = L \begin{bmatrix} M_1^{-1} & 0 & 0 \\ 0 & M_2^{-1} & 0 \\ 0 & 0 & S^{-1} \end{bmatrix} L^T$$

where

$$L = \begin{bmatrix} M_1 & 0 & 0 \\ 0 & M_2 & 0 \\ B_1^T & B_2^T & S \end{bmatrix}$$

then

$$M = \begin{bmatrix} M_1 & 0 & B_1 \\ 0 & M_2 & B_2 \\ B_1^T & B_2^T & S_* \end{bmatrix} \qquad (10.3.9)$$

with $S_* = S + B_1^T M_1^{-1} B_1 + B_2^T M_2^{-1} B_2$. Let us consider how we might choose the block parameters M_1, M_2, and S so as to produce an effective preconditioner.

If we compare (10.3.9) with the $p = 2$ version of (10.3.8) we see that it makes sense for M_i to approximate A_i and for S_* to approximate Q. The latter is achieved if $S \approx Q - B_1^T M_1^{-1} B_1 - B_2^T M_2^{-1} B_2$. There are several approaches to selecting S and they all address the fact that we cannot form the dense matrices $B_i M_i^{-1} B_i^T$. For example, as discussed in the previous subsection, tridiagonal approximations of the M_i^{-1} could be used. See Meurant (1989).

If the subdomains are sufficiently regular and it is feasible to solve linear systems that involve the A_i exactly (say by using a fast Poisson solver), then we can set $M_i = A_i$. It follows that $M = A + E$ where the $\text{rank}(E) = m$ with m being the number of interface unknowns. Thus, the preconditioned conjugate gradient algorithm would theoretically converge in $m + 1$ steps.

Regardless of the approximations that must be incorporated in the process, we see that there are significant opportunities for parallelism because the subdomain problems are decoupled. Indeed, the number of subdomains p is usually a function of both the problem geometry and the number of processors that are available for the computation.

10.3.5 Polynomial Preconditioners

The vector z defined by the preconditioner system $Mz = r$ should be thought of as an approximate solution to $Az = r$ insofar as M is an approximation of A. One way to obtain such an approximate solution is to

apply p steps of a stationary method $M_1 z^{(k+1)} = N_1 z^{(k)} + r$, $z^{(0)} = 0$. It follows that if $G = M_1^{-1} N_1$ then

$$z \equiv z^{(p)} = (I + G + \cdots G^{p-1}) M_1^{-1} r.$$

Thus, if $M^{-1} = (I + G + \cdots G^{p-1}) M_1^{-1}$ then $Mz = r$ and we can think of M as a preconditioner. Of course, it is important that M be symmetric positive definite and this constrains the choice of M_1, N_1, and p. Because M is a polynomial in G it is referred to as a *polynomial preconditioner.* This type of preconditioner is attractive from the vector/parallel point of view and has therefore attracted considerable attention.

10.3.6 Another Perspective

The polynomial preconditioner discussion points to an important connection between the classical iterations and the preconditioned conjugate gradient algorithm. Many iterative methods have as their basic step

$$x_k = x_{k-2} + \omega_k(\gamma_k z_{k-1} + x_{k-1} - x_{k-2}) \tag{10.3.10}$$

where $M z_{k-1} = r_{k-1} = b - A x_{k-1}$. For example, if we set $\omega_k = 1$, and $\gamma_k = 1$, then

$$x_k = M^{-1}(b - A x_{k-1}) + x_{k-1},$$

i.e., $M x_k = N x_{k-1} + b$, where $A = M - N$. Thus, the Jacobi, Gauss-Seidel, SOR, and SSOR methods of §10.1 have the form (10.3.10). So also does the Chebyshev semi-iterative method (10.1.12).

Following Concus, Golub, and O'Leary (1976), it is also possible to organize Algorithm 10.3.1 with a central step of the form (10.3.10):

$x_{-1} = 0$; $x_0 =$ initial guess; $k = 0$; $r_0 = b - A x_0$
while $r_k \neq 0$
    $k = k + 1$
    Solve $M z_{k-1} = r_{k-1}$ for z_{k-1}.
    $\gamma_{k-1} = z_{k-1}^T M z_{k-1} / z_{k-1}^T A z_{k-1}$
    if $k = 1$ (10.3.11)
        $\omega_1 = 1$
    else
        $\omega_k = \left(1 - \frac{\gamma_{k-1}}{\gamma_{k-2}} \frac{z_{k-1}^T M z_{k-1}}{z_{k-2}^T M z_{k-2}} \frac{1}{\omega_{k-1}}\right)^{-1}$
    end
    $x_k = x_{k-2} + \omega_k(\gamma_{k-1} z_{k-1} + x_{k-1} - x_{k-2})$
    $r_k = b - A x_k$
end
$x = x_n$

Thus, we can think of the scalars ω_k and γ_k in (10.3.11) as acceleration parameters that can be chosen to speed the convergence of the iteration $Mx_k = Nx_{k-1} + b$. Hence, any iterative method based on the splitting $A = M - N$ can be accelerated by the conjugate gradient algorithm as long as M (the preconditioner) is symmetric and positive definite.

Problems

P10.3.1 Detail an incomplete factorization procedure that is based on gaxpy Cholesky, i.e., Algorithm 4.2.1.

P10.3.2 How many n-vectors of storage is required by a practical implementation of Algorithm 10.3.1? Ignore workspaces that may be required when $Mz = r$ is solved.

Notes and References for Sec. 10.3

Our discussion of the preconditioned conjugate gradient is drawn from several sources including

P. Concus, G.H. Golub, and D.P. O'Leary (1976). " A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations," in *Sparse Matrix Computations* , ed. J.R. Bunch and D.J. Rose, Academic Press, New York.

G.H. Golub and G. Meurant (1983). *Résolution Numérique des Grandes Systèmes Linéaires,* Collection de la Direction des Etudes et Recherches de l'Electricité de France, vol. 49, Eyolles, Paris.

O. Axelsson (1985). "A Survey of Preconditioned Iterative Methods for Linear Systems of Equations," *BIT 25,* 166–187.

P. Concus, G.H. Golub, and G. Meurant (1985). "Block Preconditioning for the Conjugate Gradient Method," *SIAM J. Sci. and Stat. Comp. 6,* 220–252.

O. Axelsson and G. Lindskog (1986). "On the Rate of Convergence of the Preconditioned Conjugate Gradient Method," *Numer. Math. 48,* 499–523.

Incomplete factorization ideas are detailed in

J.A. Meijerink and H.A. Van der vorst (1977). "An Iterative Solution Method for Linear Equation Systems of Which the Coefficient Matrix is a Symmetric M-Matrix," *Math. Comp. 31,* 148–62.

T.A. Mantueffel (1979). " Shifted Incomplete Cholesky Factorization," in *Sparse Matrix Proceedings* , 1978, ed. I.S. Duff and G.W. Stewart, SIAM Publications, Philadelphia, PA.

T.F. Chan, K.R. Jackson, and B. Zhu (1983). "Alternating Direction Incomplete Factorizations," *SIAM J. Numer. Anal. 20,* 239–257.

G. Roderigue and D. Wolitzer (1984). "Preconditioning by Incomplete Block Cyclic Reduction," *Math. Comp. 42,* 549–566.

O. Axelsson (1985). "Incomplete Block Matrix Factorization Preconditioning Methods. The Ultimate Answer?", *J. Comput. Appl. Math. 12&13,* 3–18.

O. Axelsson (1986). "A General Incomplete Block Matrix Factorization Method," *Lin. Alg. Appl. 74,* 179–190.

H. Elman (1986). "A Stability Analysis of Incomplete LU Factorization," *Math. Comp. 47,* 191–218.

T. Chan (1991). "Fourier Analysis of Relaxed Incomplete Factorization Preconditioners," *SIAM J. Sci. Statist. Comput. 12,* 668–680.

Y. Notay (1992). "On the Robustness of Modified Incomplete Factorization Methods," *J. Comput. Math. 40*, 121–141.

For information on domain decomposition and other "pde driven" preconditioning ideas, see

J.H. Bramble, J.E. Pasciak, and A.H. Schatz (1986). "The construction of Preconditioners for Elliptic Problems by Substructuring I," *Math. Comp. 47*, 103–134.
J.H. Bramble, J.E. Pasciak, and A.H. Schatz (1986). "The construction of Preconditioners for Elliptic Problems by Substructuring II," *Math. Comp. 49*, 1–17.
G. Meurant (1989). "Domain Decomposition Methods for Partial Differential Equations on Parallel Computers," to appear *Int'l J. Supercomputing Applications*.
W.D. Gropp and D.E. Keyes (1992). "Domain Decomposition with Local Mesh Refinement," *SIAM J. Sci. Statist. Comput. 13*, 967–993.
D.E. Keyes, T.F. Chan, G. Meurant, J.S. Scroggs, and R.G. Voigt (eds) (1992). *Domain Decomposition Methods for Partial Differential Equations*, SIAM Publications, Philadelphia, PA.
M. Mu. (1995). "A New family of Preconditioners for Domain Decomposition," *SIAM J. Sci. Comp. 16*, 289–306.

Various aspects of polynomial preconditioners are discussed in

O.G. Johnson, C.A. Micchelli, and G. Paul (1983). "Polynomial Preconditioners for Conjugate Gradient Calculations," *SIAM J. Numer. Anal. 20*, 362–376.
S.C. Eisenstat (1984). "Efficient Implementation of a Class of Preconditioned Conjugate Gradient Methods," *SIAM J. Sci. and Stat. Computing 2*, 1–4.
Y. Saad (1985). "Practical Use of Polynomial Preconditionings for the Conjugate Gradient Method," *SIAM J. Sci. and Stat. Comp. 6*, 865–882.
L. Adams (1985). "m-step Preconditioned Congugate Gradient Methods," *SIAM J. Sci. and Stat. Comp. 6*, 452 463.
S.F. Ashby (1987). "Polynomial Preconditioning for Conjugate Gradient Methods," Ph.D. Thesis, Dept. of Computer Science, University of Illinois.
S. Ashby, T. Manteuffel, and P. Saylor (1989). "Adaptive Polynomial Preconditioning for Hermitian Indefinite Linear Systems," *BIT 29*, 583–609.
R.W. Freund (1990). "On Conjugate Gradient Type Methods and Polynomial Preconditioners for a Class of Complex Non-Hermitian Matrices," *Numer. Math. 57*, 285 312.
S. Ashby, T. Manteuffel, and J. Otto (1992). "A Comparison of Adaptive Chebyshev and Least Squares Polynomial Preconditioning for Hermitian Positive Definite Linear Systems," *SIAM J. Sci. Stat. Comp. 13*, 1–29.

Numerous vector/parallel implementations of the cg method have been developed. See

P.F. Dubois, A. Greenbaum, and G.H. Rodrigue (1979). "Approximating the Inverse of a Matrix for Use on Iterative Algorithms on Vector Processors," *Computing 22*, 257–268.
H.A. Van der Vorst (1982). "A Vectorizable Variant of Some ICCG Methods," *SIAM J. Sci. and Stat. Comp. 3*, 350–356.
G. Meurant (1984). "The Block Preconditioned Conjugate Gradient Method on Vector Computers," *BIT 24*, 623–633.
T. Jordan (1984). "Conjugate Gradient Preconditioners for Vector and Parallel Processors," in G. Birkoff and A. Schoenstadt (eds), *Proceedings of the Conference on Elliptic Problem Solvers*, Academic Press, NY.
H.A. Van der Vorst (1986). "The Performance of Fortran Implementations for Preconditioned Conjugate Gradients on Vector Computers," *Parallel Computing 3*, 49–58.
M.K. Seager (1986). "Parallelizing Conjugate Gradient for the Cray X-MP," *Parallel Computing 3*, 35–47.

O. Axelsson and B. Polman (1986). "On Approximate Factorization Methods for Block Matrices Suitable for Vector and Parallel Processors," *Lin. Alg. and Its Applic. 77*, 3–26.

D.P. O'Leary (1987). "Parallel Implementation of the Block Conjugate Gradient Algorithm," *Parallel Computers 5*, 127–140.

R. Melhem(1987). "Toward Efficient Implementation of Preconditioned Conjugate Gradient Methods on Vector Supercomputers," *Int'l J. Supercomputing Applications 1*, 70–98.

E.L. Poole and J.M. Ortega (1987). "Multicolor ICCG Methods for Vector Computers," *SIAM J. Numer. Anal. 24*, 1394–1418.

C.C. Ashcraft and R. Grimes (1988). "On Vectorizing Incomplete Factorization and SSOR Preconditioners," *SIAM J. Sci. and Stat. Comp. 9*, 122 151.

U. Meier and A. Sameh (1988). "The Behavior of Conjugate Gradient Agorithms on a Multivector Processor with a Hierarchical Memory," *J. Comput. Appl. Math. 24*, 13–32.

W.D. Gropp and D.E. Keyes (1988). "Complexity of Parallel Implementation of Domain Decomposition Techniques for Elliptic Partial Differential Equations," *SIAM J. Sci. and Stat. Comp. 9*, 312–326.

H. Van Der Vorst (1989). "High Performance Preconditioning," *SIAM J. Sci. and Stat. Comp. 10*, 1174–1185.

H. Elman (1989). "Approximate Schur Complement Preconditioners on Serial and Parallel Computers," *SIAM J. Sci. Stat. Comput. 10*, 581–605.

O. Axelsson and V. Eijkhout (1989). "Vectorizable Preconditioners for Elliptic Difference Equations in Three Space Dimensions," *J. Comput. Appl. Math. 27*, 299–321.

S.L. Johnsson and K. Mathur (1989). "Experience with the Conjugate Gradient Method for Stress Analysis on a Data Parallel Supercomputer," *International Journal on Numerical Methods in Engineering 27*, 523–546.

L. Mansfield (1991). "Damped Jacobi Preconditioning and Coarse Grid Deflation for Conjugate Gradient Iteration on Parallel Computers," *SIAM J. Sci. and Stat. Comp. 12*, 1314–1323.

V. Eijkhout (1991). "Analysis of Parallel Incomplete Point Factorizations," *Lin. Alg. and Its Applic. 154–156*, 723–740.

S. Doi (1991). "On Parallelism and Convergence of Incomplete LU Factorizations," *Appl. Numer. Math. 7*, 417–436.

Preconditioners for large Toeplitz systems are discussed in

G. Strang (1986). "A Proposal for Toeplitz Matrix Calculations," *Stud. Appl. Math. 74*, 171–176.

T.F. Chan (1988). "An Optimal Circulant Preconditioner for Toeplitz Systems," *SIAM. J. Sci. Stat. Comp. 9*, 766–771.

R.H. Chan (1989). "The Spectrum of a Family of Circulant Preconditioned Toeplitz Systems," *SIAM J. Num. Anal. 26*, 503–506.

R.H. Chan (1991). "Preconditioners for Toeplitz Systems with Nonnegative Generating Functions," *IMA J. Num. Anal. 11*, 333-345.

T. Huckle (1992). "Circulant and Skewcirculant Matrices for Solving Toeplitz Matrix Problems," *SIAM J. Matrix Anal. Appl. 13*, 767–777.

T. Huckle (1992). "A Note on Skew-Circulant Preconditioners for Elliptic Problems," *Numerical Algorithms 2*, 279–286.

R.H. Chan, J.G. Nagy, and R.J. Plemmons (1993). "FFT based Preconditioners for Toeplitz Block Least Squares Problems," *SIAM J. Num. Anal. 30*, 1740–1768.

M. Hanke and J.G. Nagy (1994). "Toeplitz Approximate Inverse Preconditioner for Banded Toeplitz Matrices," *Numerical Algorithms 7*, 183–199.

R.H. Chan, J.G. Nagy, and R.J. Plemmons (1994). "Circulant Preconditioned Toeplitz Least Squares Iterations," *SIAM J. Matrix Anal. Appl. 15*, 80–97.

T.F. Chan and J.A. Olkin (1994). "Circulant Preconditioners for Toeplitz Block Matrices," *Numerical Algorithms 6*, 89–101.

Finally, we offer an assortment of references concerned with the practical application of the cg method:

J.K. Reid (1972). "The Use of Conjugate Gradients for Systems of Linear Equations Possessing Property A," *SIAM J. Num. Anal. 9*, 325–32.

D.P. O'Leary (1980). "The Block Conjugate Gradient Algorithm and Related Methods," *Lin. Alg. and Its Applic. 29*, 293–322.

R.C. Chin, T.A. Manteuffel, and J. de Pillis (1984). "ADI as a Preconditioning for Solving the Convection-Diffusion Equation," *SIAM J. Sci. and Stat. Comp. 5*, 281–299.

I. Duff and G. Meurant (1989). "The Effect of Ordering on Preconditioned Conjugate Gradients," *BIT 29*, 635–657.

A. Greenbaum and G. Rodrigue (1989). "Optimal Preconditioners of a Given Sparsity Pattern," *BIT 29*, 610–634.

O. Axelsson and P. Vassilevski (1989). "Algebraic Multilevel Preconditioning Methods I," *Numer. Math. 56*, 157–177.

O. Axelsson and P. Vassilevski (1990). "Algebraic Multilevel Preconditioning Methods II," *SIAM J. Numer. Anal. 27*. 1569–1590.

M. Hanke and M. Neumann (1990). "Preconditionings and Splittings for Rectangular Systems," *Numer. Math. 57*, 85–96.

A. Greenbaum (1992). "Diagonal Scalings of the Laplacian as Preconditioners for Other Elliptic Differential Operators," *SIAM J. Matrix Anal. Appl. 13*, 826–846.

P.E. Gill, W. Murray, D.B. Ponceleón, and M.A. Saunders (1992). "Preconditioners for Indefinite Systems Arising in Optimization," *SIAM J. Matrix Anal. Appl. 13*, 292–311.

G. Meurant (1992). "A Review on the Inverse of Symmetric Tridiagonal and Block Tridiagonal Matrices," *SIAM J. Matrix Anal. Appl. 13*, 707 728.

S. Holmgren and K. Otto (1992). "Iterative Solution Methods and Preconditioners for Block-Tridiagonal Systems of Equations," *SIAM J. Matrix Anal. Appl. 13*, 863–886.

S.A. Vavasis (1992). "Preconditioning for Boundary Integral Equations," *SIAM J. Matrix Anal. Appl. 13*, 905–925.

P. Joly and G. Meurant (1993). "Complex Conjugate Gradient Methods," *Numerical Algorithms 4*, 379–406.

X.-C. Cai and O. Widlund (1993). "Multiplicative Schwarz Algorithms for Some Nonsymmetric and Indefinite Problems," *SIAM J. Numer. Anal. 30*, 936–952.

10.4 Other Krylov Subspace Methods

The conjugate gradient method presented over the previous two sections is applicable to symmetric positive definite systems. The MINRES and SYMMLQ variants developed in §9.3.2 in connection with the symmetric Lanczos process can handle symmetric indefinite systems. Now we push the generalizations even further in pursuit of iterative methods that are applicable to unsymmetric systems.

The discussion is patterned after the survey article by Freund, Golub, and Nachtigal (1992) and Chapter 9 of Golub and Ortega (1993). We focus on cg-type algorithms that involve optimization over Krylov spaces.

Bear in mind that there is a large gap between our algorithmic specifications and production software. A good place to build an appreciation for this point is the *Templates* book by Barrett *et al* (1993). The book by Saad (1996) is also highly recommended

10.4.1 Normal Equation Approaches

The method of normal equations for the least squares problem is appealing because it allows us to use simple "Cholesky technology" instead of more complicated methods that involve orthogonalization. Likewise, in the unsymmetric $Ax = b$ problem it is tempting to solve the equivalent symmetric positive definite system

$$A^T Ax = A^T b$$

using existing conjugate gradient technology. Indeed, if we make the substitution $A \leftarrow A^T A$ in Algorithm 10.2.1 and note that a normal equation residual $A^T b - A^T Ax_k$ is A^T times the "true" residual $b - Ax_k$, then we obtain the Conjugate Gradient Normal Equation Residual method:

Algorithm 10.4.1 [CGNR] If $A \in \mathbb{R}^{n \times n}$ is nonsingular, $b \in \mathbb{R}^n$, and $x_0 \in \mathbb{R}^n$ is an initial guess ($Ax_0 \approx b$), then the following algorithm computes $x \in \mathbb{R}^n$ so $Ax = b$.

$k = 0$
$r_0 = b - Ax_0$
while $r_k \neq 0$
  $k = k + 1$
  if $k = 1$
    $p_1 = A^T r_0$
  else
    $\beta_k = (A^T r_{k-1})^T (A^T r_{k-1}) / (A^T r_{k-2})^T (A^T r_{k-2})$
    $p_k = A^T r_{k-1} + \beta_k p_{k-1}$
  end
  $\alpha_k = (A^T r_{k-1})^T (A^T r_{k-1}) / (Ap_k)^T (Ap_k)$
  $x_k = x_{k-1} + \alpha_k p_k$
  $r_k = r_{k-1} - \alpha_k Ap_k$
end
$x = x_k$

Another way to make an unsymmetric $Ax = b$ problem "cg-friendly" is to work with the system

$$AA^T y = b \qquad x = A^T y.$$

In "y space" the cg algorithm takes on the following form:

```
k = 0
y_0 = initial guess (AA^T y_0 = b)
r_0 = b - AA^T y_0
while r_k ≠ 0
      k = k + 1
      if k = 1
            p_1 = r_0
      else
            β_k = r_{k-1}^T r_{k-1} / r_{k-2}^T r_{k-2}
            p_k = r_{k-1} + β_k p_{k-1}
      end
      α_k = r_{k-1}^T r_{k-1} / p_k^T AA^T p_k
      y_k = y_{k-1} + α_k p_k
      r_k = r_{k-1} - α_k AA^T p_k
end
y = y_k
```

Making the substitutions $x_k \leftarrow A^T y_k$ and $p_k \leftarrow A^T p_k$ and simplifying we obtain the Conjugate Gradient Normal Equation Error method:

Algorithm 10.4.2 [CGNE] If $A \in \mathbb{R}^{n\times n}$ is nonsingular, $b \in \mathbb{R}^n$, and $x_0 \in \mathbb{R}^n$ is an initial guess ($Ax_0 \approx b$), then the following algorithm computes $x \in \mathbb{R}^n$ so $Ax = b$.

```
k = 0
r_0 = b - Ax_0
while r_k ≠ 0
      k = k + 1
      if k = 1
            p_1 = A^T r_0
      else
            β_k = r_{k-1}^T r_{k-1} / r_{k-2}^T r_{k-2}
            p_k = A^T r_{k-1} + β_k p_{k-1}
      end
      α_k = r_{k-1}^T r_{k-1} / p_k^T p_k
      x_k = x_{k-1} + α_k p_k
      r_k = r_{k-1} - α_k A p_k
end
x = x_k
```

In general these two normal equation approaches are handicapped by the squaring of the condition number. (Recall Theorem 10.2.6.) However, there are some occasions where they are effective and we refer the reader to Freund, Golub and Nachtigal (1991).

10.4.2 A Note on Objective Functions

Based on what we know about the cg method, the CGNR iterate x_k minimizes

$$\phi_1(x) = \frac{1}{2}x^T(A^TA)x - x^TA^Tb$$

over the set

$$\mathcal{S}_k^{(CGNR)} = x_0 + \mathcal{K}(A^TA, r_0, k).$$

It is easy to show that

$$\frac{1}{2}\| b - Ax \|_2^2 \;=\; \phi_1(x) \;+\; \frac{1}{2}b^Tb$$

and so x_k minimizes the residual $\| b - Ax \|_2$ over $\mathcal{S}_k^{(CGNR)}$. The "R" in "CGNR" is there because of the residual-based optimization.

On the other hand, the CGNE (implicit) iterate y_k minimizes

$$\phi_2(y) = \frac{1}{2}y^T(AA^T)y - y^Tb$$

over the set $y_0 + \mathcal{K}(AA^T, b - AA^Ty_0, k)$. With the change of variable $x = A^Ty$ it can be shown that x_k minimizes

$$\frac{1}{2}x^Tx - x^TA^{-1}b = \frac{1}{2}\| x \;-\; A^{-1}b \|_2^2 \;+\; \frac{1}{2}\| A^{-1}b \|_2^2$$

over

$$\mathcal{S}_k^{(CGNE)} \;=\; x_0 + \mathcal{K}(A^TA, A^Tr_0, k). \qquad (10.4.1)$$

Thus CGNE minimizes the error at each step and that explains the "E" in "CGNE".

10.4.3 The Conjugate Residual Method

Recall that if A is symmetric positive definite, then it has a symmetric positive definite square root $A^{1/2}$. (See §4.2.10.) Note that in this case $Ax = b$ and

$$A^{1/2}x = A^{-1/2}b$$

are equivalent and that the former is the normal equation version of the latter. If we apply CGNR to this square root system and simplify the results, then we obtain

Algorithm 10.2.3 [Conjugate Residuals] If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $b \in \mathbb{R}^n$, and $x_0 \in \mathbb{R}^n$ is an initial guess ($Ax_0 \approx b$), then the following algorithm computes $x \in \mathbb{R}^n$ so $Ax = b$.

```
k = 0
r_0 = b - A x_0
while r_k ≠ 0
      k = k + 1
      if k = 1
            p_1 = r_0
      else
            β_k = r_{k-1}^T A r_{k-1} / r_{k-2}^T A r_{k-2}
            A p_k = A r_{k-1} + β_k A p_{k-1}
      end
      α_k = r_{k-1}^T A r_{k-1} / (A p_k)^T (A p_k)
      x_k = x_{k-1} + α_k p_k
      r_k = r_{k-1} - α_k A p_k
end
x = x_k
```

It follows from our comments about CGNR that $\| A^{-1/2}(b - Ax) \|_2$ is minimized over the set $x_0 + \mathcal{K}(A, r_0, k)$ during the kth iteration

10.4.4 GMRES

In §9.3.2 we briefly discussed the Lanczos-based MINRES method for symmetric, possibly indefinite, $Ax = b$ problems. In that method the iterate x_k minimizes $\| b - Ax \|_2$ over the set

$$\mathcal{S}_k = x_0 + \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\} = x_0 + \mathcal{K}(A, r_0, k) \qquad (10.4.2)$$

The key idea behind the algorithm is to express x_k in terms of the Lanczos vectors $q_1, q_2, \ldots, q_k$ which span $\mathcal{K}(A, r_0, k)$ if q_1 is a multiple of the initial residual $r_0 = b - Ax_0$.

In the Generalized Minimum Residual (GMRES) method of Saad and Schultz (1986) the same approach is taken except that the iterates are expressed in terms of Arnoldi vectors instead of Lanczos vectors in order to handle unsymmetric A. After k steps of the Arnoldi iteration (9.4.1) we have the factorization

$$AQ_k = Q_{k+1}\tilde{H}_k \qquad (10.4.3)$$

where the columns of $Q_{k+1} = [\, Q_k \; q_{k+1} \,]$ are the orthonormal Arnoldi vectors and

$$\tilde{H}_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2k} \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & h_{k,k-1} & h_{kk} \\ 0 & \cdots & \cdots & 0 & h_{k+1,k} \end{bmatrix} \in \mathbb{R}^{k+1 \times k}$$

is upper Hessenberg. In the kth step of GMRES, $\| b - Ax_k \|_2$ is minimized subject to the constraint that x_k has the form $x_k = x_0 + Q_k y_k$ for some $y_k \in \mathbb{R}^k$. If $q_1 = r_0/\rho_0$ where $\rho_0 = \| r_0 \|_2$, then it follows that

$$\begin{aligned} \| b - A(x_0 + Q_k y_k) \|_2 &= \| r_0 - AQ_k y_k \|_2 \\ &= \| r_0 - Q_{k+1} \tilde{H}_k y_k \|_2 \\ &= \| \rho_0 e_1 - \tilde{H}_k y_k \|_2. \end{aligned}$$

Thus, y_k is the solution to a $(k+1)$-by-k least squares problem and the GMRES iterate is given by $x_k = x_0 + Q_k y_k$.

Algorithm 10.4.4 [GMRES] If $A \in \mathbb{R}^{n\times n}$ is nonsingular, $b \in \mathbb{R}^n$, and $x_0 \in \mathbb{R}^n$ is an initial guess $(Ax_0 \approx b)$, then the following algorithm computes $x \in \mathbb{R}^n$ so $Ax = b$.

$r_0 = b - Ax_0$
$h_{10} = \| r_0 \|_2$
$k = 0$
while $(h_{k+1,k} > 0)$
  $q_{k+1} = r_k/h_{k+1,k}$
  $k = k + 1$
  $r_k = Aq_k$
  for $i = 1{:}k$
    $h_{ik} = q_i^T r_k$
    $r_k = r_k - h_{ik} q_i$
  end
  $h_{k+1,k} = \| r_k \|_2$
  $x_k = x_0 + Q_k y_k$ where $\| h_{10} e_1 - \tilde{H}_k y_k \|_2 = \min$
end
$x = x_k$

It is easy to verify that

$$\| b - Ax_k \|_2 = h_{k+1,k}$$

The upper Hessenberg least square problem can be efficiently solved using Givens rotations. In practice there is no need to form x_k until one is happy with its residual.

The main problem with "unlimited GMRES" is that the kth iteration involves $O(kn)$ flops. Thus like Arnoldi, a practical GMRES implementation requires a restart strategy to avoid excessive amounts of computation and memory traffic. For example, if at most m steps are tolerable, then x_m can be used as the initial vector for the next GMRES sequence.

10.4.5 Preconditioning

Preconditioning is the other key to making GMRES effective. Analogous to the development of the preconditioned conjugate gradient method in §10.3, we obtain a nonsingular matrix $M = M_1M_2$ that approximates A in some sense and then apply GMRES to the system $\tilde{A}\tilde{x} = \tilde{b}$ where $\tilde{A} = M_1^{-1}AM_2^{-1}$, $\tilde{b} = M_1^{-1}b$, and $\tilde{x} = M_2x$. If we write down the GMRES iteration for the tilde system and manipulate the equations to restore the original variables, then the resulting iteration requires the solution of linear systems that involve the preconditioner M. Thus, the act of finding a good preconditioner $M = M_1M_2$ is the act of making $\tilde{A} = M_1^{-1}AM_2^{-1}$ look as much as possible like the identity subject to the constraint that linear systems with M are easy to solve.

10.4.6 The Biconjugate Gradient Method

Just as Arnoldi underwrites GMRES, the unsymmetric Lanczos process underwrites the Biconjugate gradient (BiCG) method. The starting point in the development of BiCG is to go back to the Lanczos derivation of the conjugate gradient method in §9.3.1. In terms of Lanczos vectors, the kth cg iterate is given by $x_k = x_0 + Q_ky_k$ where Q_k is the matrix of Lanczos vectors, $T_k = Q_k^TAQ_k$ is tridiagonal, and y_k solves $T_ky_k = Q_k^Tr_0$. Note that

$$Q_k^T(b - Ax_k) = Q_k^T(r_0 - AQ_ky_k) = 0.$$

Thus, we can characterize x_k by insisting that it come from $x_0 + \mathcal{K}(A, r_0, k)$ and that it produce a residual that is orthogonal to a given subspace, say $\mathcal{K}(A, r_0, k)$.

In the unsymmetric case we can extend this notion by producing a sequence of iterates $\{x_k\}$ with the property that x_k belongs to $x_0 + \mathcal{K}(A, r_0, k)$ and produces a residual that is orthogonal to $\mathcal{K}(A^T, s_0, k)$ for some $s_0 \in \mathbb{R}^n$. Simplifications occur if the unsymmetric Lanczos process is used to generate bases for the two involved Krylov spaces. In particular, after k steps of the unsymmetric Lanczos algorithm (9.4.7) we have $Q_k, P_k \in \mathbb{R}^{n\times k}$ such that $P_k^TQ_k = I_k$ and a tridiagonal matrix $T_k = P_k^TAQ_k$ such that

$$\begin{aligned} AQ_k &= Q_kT_k + r_ke_k^T \qquad P_k^Tr_k = 0 \\ A^TP_k &= P_kT_k^T + s_ke_k^T \qquad Q_k^Ts_k = 0 \end{aligned} \tag{10.4.4}$$

In BiCG we set $x_k = x_0 + Q_ky_k$ where $T_ky_k = Q_k^Tr_0$. Note that the *Galerkin condition*

$$P_k^T(b - Ax_k) = P_k^T(r_0 - AQ_ky_k) = 0$$

holds.

As might be expected, it is possible to develop recursions so that x_k can be computed as a simple combination of x_{k-1} and q_{k-1}, instead of as a linear combination of all the previous q-vectors.

The BiCG method is subject to serious breakdown because of its dependence on the unsymmetric Lanczos process. However, by relying on a look-ahead Lanczos procedure it is possible to overcome some of these difficulties.

10.4.7 QMR

Another iteration that runs off of the unsymmetric Lanczos process is the quasi-minimum residual (QMR) method of Freund and Nachtigal (1991). As in BiCG the kth iterate has the form $x_k = x_0 + Q_k y_k$. It is easy to show that after k steps in (9.4.7) we have the factorization

$$AQ_k = Q_{k+1}\tilde{T}_k$$

where $\tilde{T}_k \in \mathbb{R}^{k+1 \times k}$ is tridiagonal. It follows that if $q_1 = \rho(b - Ax_0)$, then

$$\begin{aligned} b - Ax_k &= b - A(x_0 + Q_k y_k) \\ &= r_0 - AQ_k y_k \\ &= r_0 - Q_{k+1}\tilde{T}_k y_k \\ &= Q_{k+1}(\rho e_1 - \tilde{T}_k y_k). \end{aligned}$$

If y_k is chosen to minimize the 2-norm of this vector, then in exact arithmetic $x_0 + Q_k y_k$ defines the GMRES iterate. In QMR, y_k is chosen to minimize $\| \rho e_1 - \tilde{T}_k y_k \|_2$.

10.4.8 Summary

The methods that we have presented do not submit to a linear ranking. The choice of a technique is complicated and depends on a host of factors. A particularly cogent assessment of the major algorithms is given in Barrett *et al* (1993).

Problems

P10.4.1 Analogous to (10.2.16), develop efficient implementations of the CGNR, CGNE, Conjugate residual methods.

P10.4.2 Establish the mathematical equivalence of the CGNR and the LSQR method outlined in §9.3.4.

P10.4.3 Prove (10.4.3).

P10.4.4 Develop an efficient preconditioned GMRES implementation. Proceeding as we did in §10.3 for preconditioned conjugate gradient method. (See (10.3.2) and (10.3.3) in particular.)

P10.4.5 Prove that the GMRES least squares problem has full rank.

Notes and References for Sec. 10.4

The following papers serve as excellent introductions to the world of unsymmetric iteration:

S. Eisenstat, H. Elman, and M. Schultz (1983). "Variational Iterative Methods for Nonsymmetric Systems of Equations," *SIAM J. Num. Anal. 20,* 345–357.

R.W. Freund, G.H. Golub, and N. Nachtigal (1992). "Iterative Solution of Linear Systems," *Acta Numerica 1,* 57–100.

N. Nachtigal, S. Reddy, and L. Trefethen (1992). "How Fast Are Nonsymmetric Matrix Iterations," *SIAM J. Matrix Anal. Appl. 13,* 778–795.

A. Greenbaum and L.N. Trefethen (1994). "GMRES/CR and Arnoldi/Lanczos as Matrix Approximation Problems," *SIAM J. Sci. Comp. 15,* 359–368.

Krylov space methods and analysis are featured in the following papers:

W.E. Arnoldi (1951). "The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem," *Quart. Appl. Math. 9,* 17–29.

Y. Saad (1981). "Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems," *Math. Comp. 37,* 105–126.

Y. Saad (1984). "Practical Use of Some Krylov Subspace Methods for Solving Indefinite and Nonsymmetric Linear Systems," *SIAM J. Sci. and Stat. Comp. 5,* 203–228.

Y. Saad (1989). "Krylov Subspace Methods on Supercomputers," *SIAM J. Sci. and Stat. Comp. 10,* 1200–1322.

C.-M. Huang and D.P. O'Leary (1993). "A Krylov Multisplitting Algorithm for Solving Linear Systems of Equations," *Lin. Alg. and Its Applic. 194,* 9–29.

C.C. Paige, B.N. Parlett,and H.A. Van Der Vorst (1995). "Approximate Solutions and Eigenvalue Bounds from Krylov Subspaces," *Numer. Linear Algebra with Applic. 2,* 115–134.

References for the GMRES method include

Y. Saad and M. Schultz (1986). "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Scientific and Stat. Comp. 7,* 856–869.

H.F. Walker (1988). "Implementation of the GMRES Method Using Householder Transformations," *SIAM J. Sci. Stat. Comp. 9,* 152–163.

C. Vuik and H.A. van der Vorst (1992). "A Comparison of Some GMRES-like Methods," *Lin. Alg. and Its Applic. 160,* 131–162.

N. Nachtigal, L. Reichel, and L. Trefethen (1992). "A Hybrid GMRES Algorithm for Nonsymmetric Linear Systems," *SIAM J. Matrix Anal. Appl. 13,* 796–825.

Y. Saad (1993). "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM J. Sci. Comput. 14,* 461–469.

Z. Bai, D. Hu, and L. Reichel (1994). "A Newton Basis GMRES Implementation," *IMA J. Num. Anal. 14,* 563–581.

R.B. Morgan (1995). "A Restarted GMRES Method Augmented with Eigenvectors," *SIAM J. Matrix Anal. Applic. 16,* 1154–1171.

Preconditioning ideas for unsymmetric problems are discussed in the following papers:

Y. Saad (1988). "Preconditioning Techniques for Indefinite and Nonsymmetric Linear Systems," *J. Comput. Appl. Math. 24,* 89–105.

L. Yu. Kolotilina and A. Yu. Yeremin (1993). "Factorized Sparse Approximate Inverse Preconditioning I: Theory," *SIAM J. Matrix Anal. Applic. 14,* 45–58.

I.E. Kaporin (1994). "New Convergence Results and Preconditioning Strategies for the Conjugate Gradient Method," *Num. Lin. Alg. Applic. 1,* 179–210.

L. Yu. Kolotilina and A. Yu. Yeremin (1995). "Factorized Sparse Approximate Inverse Preconditioning II: Solution of 3D FE Systems on Massively Parallel Computers," *Intern. J. High Speed Comput. 7,* 191–215.

H. Elman (1996). "Fast Nonsymmetric Iterations and Preconditioning for Navier-Stokes Equations," *SIAM J. Sci. Comput. 17,* 33–46.

M. Benzi, C.D. Meyer, and M. Tuma (1996). "A Sparse Approximate Inverse Preconditioner for the Conjugate Gradient Method," *SIAM J. Sci. Comput. 17*, to appear.

Some representative papers concerned with the development of nonsymmetric conjugate gradient procedures include

D.M. Young and K.C. Jea (1980). "Generalized Conjugate Gradient Acceleration of Nonsymmetrizable Iterative Methods," *Lin. Alg. and Its Applic. 34*, 159–94.

O. Axelsson (1980). "Conjugate Gradient Type Methods for Unsymmetric and Inconsistent Systems of Linear Equations," *Lin. Alg. and Its Applic. 29*, 1–16.

K.C. Jea and D.M. Young (1983). "On the Simplification of Generalized Conjugate Gradient Methods for Nonsymmetrizable Linear Systems," *Lin. Alg. and Its Applic. 52/53*, 399–417.

V. Faber and T. Manteuffel (1984). "Necessary and Sufficient Conditions for the Existence of a Conjugate Gradient Method," *SIAM J. Numer. Anal. 21* 352–362.

Y. Saad and M. Schultz (1985). "Conjugate Gradient-Like Algorithms for Solving Nonsymmetric Linear Systems," *Math. Comp. 44*, 417–424.

H.A. Van der Vorst (1986). "An Iterative Solution Method for Solving $f(A)x = b$ Using Krylov Subspace Information Obtained for the Symmetric Positive Definite Matrix A," *J. Comp. and App. Math. 18*, 249–263.

M.A. Saunders, H.D. Simon, and E.L. Yip (1988). "Two Conjugate Gradient-Type Methods for Unsymmetric Linear Equations," *SIAM J. Num. Anal. 25*, 927–940.

R. Freund (1992). "Conjugate Gradient-Type Methods for Linear Systems with Complex Symmetric Coefficient Matrices," *SIAM J. Sci. Statist. Comput. 13*, 425–448.

More Lanczos-based solvers are discussed in

Y. Saad (1982). "The Lanczos Biorthogonalization Algorithm and Other Oblique Projection Methods for Solving Large Unsymmetric Systems," *SIAM J. Numer. Anal. 19*, 485–506.

Y. Saad (1987). "On the Lanczos Method for Solving Symmetric Systems with Several Right Hand Sides," *Math. Comp. 48*, 651–662.

C. Brezinski and H. Sadok (1991). "Avoiding Breakdown in the CGS Algorithm," *Numer. Alg. 1*, 199–206.

C. Brezinski, M. Zaglia, and H. Sadok (1992). "A Breakdown Free Lanczos Type Algorithm for Solving Linear Systems," *Numer. Math. 63*, 29–38.

S.K. Kim and A.T. Chronopoulos (1991). "A Class of Lanczos-Like Algorithms Implemented on Parallel Computers," *Parallel Comput. 17*, 763–778.

W. Joubert (1992). "Lanczos Methods for the Solution of Nonsymmetric Systems of Linear Equations," *SIAM J. Matrix Anal. Appl. 13*, 926–943.

R.W. Freund, M. Gutknecht, and N. Nachtigal (1993). "An Implementation of the Look-Ahead Lanczos Algorithm for Non-Hermitian Matrices," *SIAM J. Sci. and Stat.Comp. 14*, 137–158.

The QMR method is detailed in the following papers

R.W. Freund and N. Nachtigal (1991). "QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems," *Numer. Math. 60*, 315–339.

R.W. Freund (1993). "A Transpose-Free Quasi-Minimum Residual Algorithm for Non-hermitian Linear System," *SIAM J. Sci. Comput. 14*, 470–482.

R.W. Freund and N.M. Nachtigal (1994). "An Implementation of the QMR Method Based on Coupled Two-term Recurrences," *SIAM J. Sci. Comp. 15*, 313 337.

The residuals in BiCG tend to display erratic behavior prompting the development of stabilizing techniques:

H. van der Vorst (1992). "BiCGSTAB: A Fast and Smoothly Converging Variant of the Bi-CG for the Solution of Nonsymmetric Linear Systems," *SIAM J. Sci. and Stat. Comp. 13*, 631–644.

M. Gutknecht (1993). "Variants of BiCBSTAB for Matrices with Complex Spectrum," *SIAM J. Sci. and Stat. Comp. 14*, 1020–1033.

G.L.G. Sleijpen and D.R. Fokkema (1993). "BICGSTAB(ℓ) for Linear Equations Involving Unsymmetric Matrices with Complex Spectrum," *Electronic Transactions on Numerical Analysis 1*, 11–32.

C. Brezinski and M. Redivo-Zaglia (1995). "Look-Ahead in BiCGSTAB and Other Product-Type Methods for Linear Systems," *BIT 35*, 169–201.

In some applications it is awkward to produce matrix-vector product code for both Ax and A^Tx. Transpose free methods are popular in this context. See

P. Sonneveld (1989). "CGS, A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems," *SIAM J. Sci. and Stat. Comp. 10*, 36–52.

G. Radicati di Brozolo and Y. Robert (1989). "Parallel Conjugate Gradient-like Algorithms for Solving Sparse Nonsymmetric Linear Systems on a Vector Multiprocessor," *Parallel Computing 11*, 233-240.

C. Brezinski and M. Redivo-Zaglia (1994). "Treatment of Near-Breakdown in the CGS Algorithms," *Numerical Algorithms 7*, 33–73.

E.M. Kasenally (1995). "GMBACK: A Generalized Minimum Backward Error Algorithm for Nonsymmetric Linear Systems," *SIAM J. Sci. Comp. 16*, 698–719.

C.C. Paige, B.N. Parlett, and H.A. van der Vorst (1995). "Approximate Solutions and Eigenvalue Bounds from Krylov Subspaces," *Num. Lin. Alg. with Applic. 2*, 115–133.

M. Hochbruck and Ch. Lubich (1996), "On Krylov Subspace Approximations to the Matrix Exponential Operator," *SIAM J. Numer. Anal.*, to appear.

M. Hochbruck and Ch. Lubich (1996), "Error Analysis of Krylov Method in a Nutshell," *SIAM J. Sci. Comput.*, to appear.

Connections between the pseudoinverse of a rectangular matrix A and the conjugate gradient method applied to A^TA are pointed out in the paper

M. Hestenes (1975). "Pseudoinverses and Conjugate Gradients," *CACM 18*, 40–43.

Chapter 11

Functions of Matrices

Computing a function $f(A)$ of an n-by-n matrix A is a frequently occurring problem in control theory and other application areas. Roughly speaking, if the scalar function $f(z)$ is defined on $\lambda(A)$, then $f(A)$ is defined by substituting "A" for "z" in the "formula" for $f(z)$. For example, if $f(z) = (1+z)/(1-z)$ and $1 \notin \lambda(A)$, then $f(A) = (I+A)(I-A)^{-1}$.

The computations get particularly interesting when the function f is transcendental. One approach in this more complicated situation is to compute an eigenvalue decomposition $A = YBY^{-1}$ and use the formula $f(A) = Yf(B)Y^{-1}$. If B is sufficiently simple, then it is often possible to calculate $f(B)$ directly. This is illustrated in §11.1 for the Jordan and Schur decompositions. Not surprisingly, reliance on the latter decomposition results in a more stable $f(A)$ procedure.

Another class of methods for the matrix function problem is to approximate the desired function $f(A)$ with an easy-to-calculate function $g(A)$. For example, g might be a truncated Taylor series approximate to f. Error bounds associated with the approximation of matrix functions are given in §11.2.

In the last section we discuss the special and very important problem of computing the matrix exponential e^A.

Before You Begin

Chapters 1, 2, 3, 7 and 8 are assumed. Within this chapter there are the following dependencies:

$$\S 11.1 \quad \rightarrow \quad \S 11.2 \quad \rightarrow \quad \S 11.3$$

Complementary references include Mirsky (1955), Gantmacher (1959), Bellman (1969), and Horn and Johnson (1991). Some Matlab functions important to this chapter are `expm`, `expm1`, `expm2`, `expm3`, `logm`, `sqrtm`, and `funm`.

11.1 Eigenvalue Methods

Given an n-by-n matrix A and a scalar function $f(z)$, there are several ways to define the *matrix function* $f(A)$. A very informal definition might be to substitute "A" for "z" in the formula for $f(z)$. For example, if $p(z) = 1 + z$ and $r(z) = (1 - (z/2))^{-1}(1 + (z/2))$ for $z \neq 2$, then it is certainly reasonable to define $p(A)$ and $r(A)$ by

$$p(A) \;=\; I + A$$

and

$$r(A) \;=\; \left(I - \frac{A}{2}\right)^{-1} \left(I + \frac{A}{2}\right) \qquad 2 \notin \lambda(A)\,.$$

"A-for-z" substitution also works for transcendental functions, i.e.,

$$e^A \;=\; \sum_{k=0}^{\infty} \frac{A^k}{k!}\,.$$

To make subsequent algorithmic developments precise, however, we need a more precise definition of $f(A)$.

11.1.1 A Definition

There are many ways to establish rigorously the notion of a matrix function. See Rinehart (1955). Perhaps the most elegant approach is in terms of a line integral. Suppose $f(z)$ is analytic inside on a closed contour Γ which encircles $\lambda(A)$. We define $f(A)$ to be the matrix

$$f(A) \;=\; \frac{1}{2\pi i} \oint_{\Gamma} f(z)(zI - A)^{-1} dz\,. \tag{11.1.1}$$

This definition is immediately recognized as a matrix version of the Cauchy integral theorem. The integral is defined on an element-by-element basis:

$$f(A) \;=\; (f_{kj}) \quad \Longrightarrow \quad f_{kj} \;=\; \frac{1}{2\pi i} \oint_{\Gamma} f(z) e_k^T (zI - A)^{-1} e_j dz\,.$$

Notice that the entries of $(zI - A)^{-1}$ are analytic on Γ and that $f(A)$ is defined whenever $f(z)$ is analytic in a neighborhood of $\lambda(A)$.

11.1.2 The Jordan Characterization

Although fairly useless from the computational point of view, the definition (11.1.1) can be used to derive more practical characterizations of $f(A)$. For example, if $f(A)$ is defined and

$$A = XBX^{-1} = X\text{diag}(B_1, \ldots, B_p)X^{-1}, \qquad B_i \in \mathbb{C}^{n_i \times n_i}$$

then it is easy to verify that

$$f(A) = Xf(B)X^{-1} = X\text{diag}(f(B_1), \ldots, f(B_p))X^{-1}. \qquad (11.1.2)$$

For the case when the B_i are Jordan blocks we obtain the following:

Theorem 11.1.1 *Let $X^{-1}AX = \text{diag}(J_1, \ldots, J_p)$ be the Jordan canonical form (JCF) of $A \in \mathbb{C}^{n \times n}$ with*

$$J_i = \begin{bmatrix} \lambda_i & 1 & \cdots & \cdots & 0 \\ 0 & \lambda_i & 1 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 & \lambda_i \end{bmatrix}$$

being an m_i-by-m_i Jordan block. If $f(z)$ is analytic on an open set containing $\lambda(A)$, then

$$f(A) = X\text{diag}(f(J_1), \ldots, f(J_p))X^{-1}$$

where

$$f(J_i) = \begin{bmatrix} f(\lambda_i) & f^{(1)}(\lambda_i) & \cdots & \cdots & \dfrac{f^{(m_i-1)}(\lambda_i)}{(m_i-1)!} \\ 0 & f(\lambda_i) & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & f^{(1)}(\lambda_i) \\ 0 & \cdots & \cdots & \cdots & f(\lambda_i) \end{bmatrix}.$$

Proof. In view of the remarks preceding the statement of the theorem, it suffices to examine $f(G)$ where

$$G = \lambda I + E \qquad E = (\delta_{i,j-1})$$

is a q-by-q Jordan block. Suppose $(zI - G)$ is nonsingular. Since

$$(zI - G)^{-1} = \sum_{k=0}^{q-1} \frac{E^k}{(z-\lambda)^{k+1}}$$

it follows from Cauchy's integral theorem that

$$f(G) = \sum_{k=0}^{q-1}\left[\frac{1}{2\pi i}\oint_{\Gamma}\frac{f(z)}{(z-\lambda)^{k+1}}dz\right]E^k = \sum_{k=0}^{q-1}\frac{f^{(k)}(\lambda)}{k!}E^k .$$

The theorem follows from the observation that $E^k = (\delta_{i,j-k})$. □

Corollary 11.1.2 *If* $A \in \mathbb{C}^{n\times n}$, $A = X\text{diag}(\lambda_1,\ldots,\lambda_n)X^{-1}$, *and* $f(A)$ *is defined, then*

$$f(A) = X\text{diag}(f(\lambda_1),\ldots,f(\lambda_n))X^{-1}.$$

Proof. The Jordan blocks are all 1-by-1. □

These results illustrate the close connection between $f(A)$ and the eigensystem of A. Unfortunately, the JCF approach to the matrix function problem has dubious computational merit unless A is diagonalizable with a well-conditioned matrix of eigenvectors. Indeed, rounding errors of order $\mathbf{u}\kappa_2(X)$ can be expected to contaminate the computed result, since a linear system involving the matrix X must be solved. The following example suggests that ill-conditioned similarity transformations should be avoided when computing a function of a matrix.

Example 11.1.1 If

$$A = \begin{bmatrix} 1+10^{-5} & 1 \\ 0 & 1-10^{-5} \end{bmatrix},$$

then any matrix of eigenvectors is a column scaled version of

$$X = \begin{bmatrix} 1 & -1 \\ 0 & 2(1-10^{-5}) \end{bmatrix}$$

and has a 2-norm condition number of order 10^5. Using a computer with machine precision $\mathbf{u} \approx 10^{-7}$ we find

$$fl[X^{-1}\text{diag}(\exp(1+10^{-5}),\exp(1-10^{-5}))X] = \begin{bmatrix} 2.718307 & 2.750000 \\ 0.000000 & 2.718254 \end{bmatrix}$$

while

$$e^A = \begin{bmatrix} 2.718309 & 2.718282 \\ 0.000000 & 2.718255 \end{bmatrix}.$$

11.1.3 A Schur Decomposition Approach

Some of the difficulties associated with the Jordan approach to the matrix function problem can be circumvented by relying upon the Schur decomposition. If $A = QTQ^H$ is the Schur decomposition of A, then

$$f(A) = Qf(T)Q^H.$$

For this to be effective, we need an algorithm for computing functions of upper triangular matrices. Unfortunately, an explicit expression for $f(T)$ is very complicated as the following theorem shows.

Theorem 11.1.3 *Let $T = (t_{ij})$ be an n-by-n upper triangular matrix with $\lambda_i = t_{ii}$ and assume $f(T)$ is defined. If $f(T) = (f_{ij})$, then $f_{ij} = 0$ if $i > j$, $f_{ij} = f(\lambda_i)$ for $i = j$, and for all $i < j$ we have*

$$f_{ij} = \sum_{(s_0,\ldots,s_k)\in S_{ij}} t_{s_0,s_1} t_{s_1,s_2} \cdots t_{s_{k-1},s_k} f[\lambda_{s_0},\ldots,\lambda_{s_k}],$$

where S_{ij} is the set of all strictly increasing sequences of integers that start at i and end at j and $f[\lambda_{s_0},\ldots,\lambda_{s_k}]$ is the kth order divided difference of f at $\{\lambda_{s_0},\ldots,\lambda_{s_k}\}$.

Proof. See Descloux (1963), Davis (1973), or Van Loan (1975). □

Computing $f(T)$ via Theorem 11.1.3 would require $O(2^n)$ flops. Fortunately, Parlett (1974) has derived an elegant recursive method for determining the strictly upper triangular portion of the matrix $F = f(T)$. It requires only $2n^3/3$ flops and can be derived from the following commutivity result:

$$FT = TF. \tag{11.1.3}$$

Indeed, by comparing (i, j) entries in this equation, we find

$$\sum_{k=i}^{j} f_{ik} t_{kj} = \sum_{k=i}^{j} t_{ik} f_{kj} \qquad j > i$$

and thus, if t_{ii} and t_{jj} are distinct,

$$f_{ij} = t_{ij} \frac{f_{jj} - f_{ii}}{t_{jj} - t_{ii}} + \sum_{k=i+1}^{j-1} \frac{t_{ik} f_{kj} - f_{ik} t_{kj}}{t_{jj} - t_{ii}}. \tag{11.1.4}$$

From this we conclude that f_{ij} is a linear combination of its neighbors to its left and below in the matrix F. For example, the entry f_{25} depends upon f_{22}, f_{23}, f_{24}, f_{55}, f_{45}, and f_{35}. Because of this, the entire upper triangular portion of F can be computed one superdiagonal at a time beginning with the diagonal, $f(t_{11}), \ldots, f(t_{nn})$. The complete procedure is as follows:

Algorithm 11.1.1 This algorithm computes the matrix function $F = f(T)$ where T is upper triangular with distinct eigenvalues and f is defined on $\lambda(T)$.

```
for i = 1:n
    f_ii = f(t_ii)
end
```

```
for p = 1:n - 1
    for i = 1:n - p
        j = i + p
        s = t_ij(f_jj - f_ii)
        for k = i + 1:j - 1
            s = s + t_ik f_kj - f_ik t_kj
        end
        f_ij = s/(t_jj - t_ii)
    end
end
```

This algorithm requires $2n^3/3$ flops. Assuming that $T = QAQ^H$ is the Schur form of A, $f(A) = QFQ^H$ where $F = f(T)$. Clearly, most of the work in computing $f(A)$ by this approach is in the computation of the Schur decomposition, unless f is extremely expensive to evaluate.

Example 11.1.2 If

$$T = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 3 & 4 \\ 0 & 0 & 5 \end{bmatrix}$$

and $f(z) = (1+z)/z$ then $F = (f_{ij}) = f(T)$ is defined by

$$\begin{aligned} f_{11} &= (1+1)/1 = 2 \\ f_{22} &= (1+3)/3 = 4/3 \\ f_{33} &= (1+5)/5 = 6/5 \\ f_{12} &= t_{12}(f_{22} - f_{11})/(t_{22} - t_{11}) = -2/3 \\ f_{23} &= t_{23}(f_{33} - f_{22})/(t_{33} - t_{22}) = -4/15 \\ f_{13} &= [t_{13}(f_{33} - f_{11}) + (t_{12}f_{23} - f_{12}t_{23})]/(t_{33} - t_{11}) = -1/15. \end{aligned}$$

11.1.4 A Block Schur Approach

If A has close or multiple eigenvalues, then Algorithm 11.1.1 leads to poor results. In this case, it is advisable to use a block version of Algorithm 11.1.1. We outline such a procedure due to Parlett (1974a). The first step is to choose Q in the Schur decomposition such that close or multiple eigenvalues are clustered in blocks $T_{11}, \ldots, T_{pp}$ along the diagonal of T. In particular, we must compute a partitioning

$$T = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1p} \\ 0 & T_{22} & \cdots & T_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{pp} \end{bmatrix} \qquad F = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1p} \\ 0 & F_{22} & \cdots & F_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F_{pp} \end{bmatrix}$$

where $\lambda(T_{ii}) \cap \lambda(T_{jj}) \neq \emptyset$, $i \neq j$. The actual determination of the block sizes can be done using the methods of §7.6.

Next, we compute the submatrices $F_{ii} = f(T_{ii})$ for $i = 1{:}p$. Since the eigenvalues of T_{ii} are presumably close, these calculations require special methods. (Some possibilities are discussed in the next two sections.) Once the diagonal blocks of F are known, the blocks in the strict upper triangle of F can be found recursively, as in the scalar case. To derive the governing equations, we equate (i, j) blocks in $FT = TF$ for $i < j$ and obtain the following generalization of (11.1.4):

$$F_{ij}T_{jj} - T_{ii}F_{ij} \;=\; T_{ij}F_{jj} - F_{ii}T_{ij} \;+\; \sum_{k=i+1}^{j-1} (T_{ik}F_{kj} - F_{ik}T_{kj})\,. \quad (11.1.5)$$

This is a linear system whose unknowns are the elements of the block F_{ij} and whose right-hand side is "known" if we compute the F_{ij} one block super-diagonal at a time. We can solve (11.1.5) using the Bartels-Stewart algorithm (Algorithm 7.6.2).

The block Schur approach described here is useful when computing real functions of real matrices. After computing the real Schur form $A = QTQ^T$, the block algorithm can be invoked in order to handle the 2-by-2 bumps along the diagonal of T.

Problems

P11.1.1 Using the definition (11.1.1) show that (a) $Af(A) = f(A)A$, (b) $f(A)$ is upper triangular if A is upper triangular, and (c) $f(A)$ is Hermitian if A is Hermitian.

P11.1.2 Rewrite Algorithm 11.1.1 so that $f(T)$ is computed column by column.

P11.1.3 Suppose $A = X\text{diag}(\lambda_i)X^{-1}$ where $X = [\, x_1, \ldots, x_n \,]$ and $X^{-1} = [\, y_1, \ldots, y_n \,]^H$. Show that if $f(A)$ is defined, then

$$f(A) \;=\; \sum_{k=1}^{n} f(\lambda_i)x_iy_i^H\,.$$

P11.1.4 Show that

$$T \;=\; \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{matrix} p \\ q \end{matrix} \quad \Rightarrow \quad f(T) \;=\; \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$$
$$\qquad\quad p \quad\; q \qquad\qquad\qquad\qquad\qquad p \quad\; q$$

where $F_{11} = f(T_{11})$ and $F_{22} = f(T_{22})$. Assume $f(T)$ is defined.

Notes and References for Sec. 11.1

The contour integral representation of $f(A)$ given in the text is useful in functional analysis because of its generality. See

N. Dunford and J. Schwartz (1958). *Linear Operators, Part I*, Interscience, New York.

As we discussed, other definitions of $f(A)$ are possible. However, for the matrix functions typically encountered in practice, all these definitions are equivalent. See

R.F. Rinehart (1955). "The Equivalence of Definitions of a Matric Function," *Amer. Math. Monthly 62*, 395–414.

Various aspects of the Jordan representation are detailed in

J.S. Frame (1964). "Matrix Functions and Applications, Part II," *IEEE Spectrum 1 (April)*, 102–8.
J.S. Frame (1964). "Matrix Functions and Applications, Part IV," *IEEE Spectrum 1 (June)*, 123–31.

The following are concerned with the Schur decomposition and its relationship to the $f(A)$ problem:

D. Davis (1973). "Explicit Functional Calculus," *Lin. Alg. and Its Applic. 6*, 193–99.
J. Descloux (1963). "Bounds for the Spectral Norm of Functions of Matrices," *Numer. Math. 5*, 185–90.
C.F. Van Loan (1975). "A Study of the Matrix Exponential," Numerical Analysis Report No. 10, Dept. of Maths., University of Manchester, England.

Algorithm 11.1.1 and the various computational difficulties that arise when it is applied to a matrix having close or repeated eigenvalues are discussed in

B.N. Parlett (1976). "A Recurrence Among the Elements of Functions of Triangular Matrices," *Lin. Alg. and Its Applic. 14*, 117–21.

A compromise between the Jordan and Schur approaches to the $f(A)$ problem results if A is reduced to block diagonal form as described in §7.6.3. See

B. Kågström (1977). "Numerical Computation of Matrix Functions," Department of Information Processing Report UMINF-58.77, University of Ümea, Sweden.

The sensitivity of matrix functions to perturbation is discussed in

C.S. Kenney and A.J. Laub (1989). "Condition Estimates for Matrix Functions," *SIAM J. Matrix Anal. Appl. 10*, 191–209.
C.S. Kenney and A.J. Laub (1994). "Small-Sample Statistical Condition Estimates for General Matrix Functions," *SIAM J. Sci. Comp. 15*, 36–61.

A theme in this chapter is that if A is nonnormal, then there is more to computing $f(A)$ than just computing $f(z)$ on $\lambda(A)$. The pseudo-eigenvalue concept is a way of understanding this phenomena. See

L.N. Trefethen (1992). "Pseudospectra of Matrices," in *Numerical Analysis 1991*, D.F. Griffiths and G.A. Watson (eds), Longman Scientific & Technical, Harlow, Essex, UK.

More details are offered in §11.3.4.

11.2 Approximation Methods

We now consider a class of methods for computing matrix functions which at first glance do not appear to involve eigenvalues. These techniques are based on the idea that if $g(z)$ approximates $f(z)$ on $\lambda(A)$, then $f(A)$ approximates $g(A)$, e.g.,

$$e^A \approx I + A + \frac{A^2}{2!} + \cdots + \frac{A^q}{q!}.$$

We begin by bounding $\| f(A) - g(A) \|$ using the Jordan and Schur matrix function representations. We follow this discussion with some comments on the evaluation of matrix polynomials.

11.2.1 A Jordan Analysis

The Jordan representation of matrix functions (Theorem 11.1.1) can be used to bound the error in an approximant $g(A)$ of $f(A)$.

Theorem 11.2.1 *Let $X^{-1}AX = \text{diag}(J_1, \ldots, J_p)$ be the JCF of $A \in \mathbb{C}^{n \times n}$ with*

$$J_i = \begin{bmatrix} \lambda_i & 1 & \cdots & \cdots & 0 \\ 0 & \lambda_i & 1 & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & \cdots & \cdots & \cdots & \lambda_i \end{bmatrix}$$

being an m_i-by-m_i Jordan block. If $f(z)$ and $g(z)$ are analytic on an open set containing $\lambda(A)$, then

$$\| f(A) - g(A) \|_2 \;\le\; \kappa_2(X) \max_{\substack{1 \le i \le p \\ 0 \le r \le m_i - 1}} m_i \frac{|f^{(r)}(\lambda_i) - g^{(r)}(\lambda_i)|}{r!}.$$

Proof. Defining $h(z) = f(z) - g(z)$ we have

$$\begin{aligned} \| f(A) - g(A) \|_2 &= \| X \text{diag}(h(J_1), \ldots, h(J_p)) X^{-1} \|_2 \\ &\le \kappa_2(X) \max_{1 \le i \le p} \| h(J_i) \|_2 . \end{aligned}$$

Using Theorem 11.1.1 and equation (2.3.8) we conclude that

$$\| h(J_i) \|_2 \le m_i \max_{0 \le r \le m_i - 1} \frac{|h^{(r)}(\lambda_i)|}{r!}$$

thereby proving the theorem. □

11.2.2 A Schur Analysis

If we rely on the Schur instead of the Jordan decomposition we obtain an alternative bound.

Theorem 11.2.2 *Let $Q^H AQ = T = \text{diag}(\lambda_i) + N$ be the Schur decomposition of $A \in \mathbb{C}^{n\times n}$, with N being the strictly upper triangular portion of T. If $f(z)$ and $g(z)$ are analytic on a closed convex set Ω whose interior contains $\lambda(A)$, then*

$$\| f(A) - g(A) \|_F \leq \sum_{r=0}^{n-1} \delta_r \frac{\| \, |N|^r \, \|_F}{r!}$$

where

$$\delta_r = \sup_{z\in\Omega} \left| f^{(r)}(z) - g^{(r)}(z) \right| .$$

Proof. Let $h(z) = f(z) - g(z)$ and set $H = (h_{ij}) = h(A)$. Let $S_{ij}^{(r)}$ denote the set of strictly increasing integer sequences $(s_0, \ldots, s_r)$ with the property that $s_0 = i$ and $s_r = j$. Notice that

$$S_{ij} = \bigcup_{r=1}^{j-i} S_{ij}^{(r)}$$

and so from Theorem 11.1.3, we obtain the following for all $i < j$:

$$h_{ij} = \sum_{r=1}^{j-1} \sum_{s\in S_{ij}^{(r)}} n_{s_0,s_1} n_{s_1,s_2} \cdots n_{s_{r-1},s_r} h\left[\lambda_{s_0}, \ldots, \lambda_{s_r}\right].$$

Now since Ω is convex and h analytic, we have

$$|h\left[\lambda_{s_0}, \ldots, \lambda_{s_r}\right]| \leq \sup_{z\in\Omega} \frac{|h^{(r)}(z)|}{r!} = \frac{\delta_r}{r!}. \qquad (11.2.1)$$

Furthermore if $|N|^r = (n_{ij}^{(r)})$ for $r \geq 1$, then it can be shown that

$$n_{ij}^{(r)} = \begin{cases} 0 & j < i + r \\ \displaystyle\sum_{s\in S_{ij}^{(r)}} \left| n_{s_0,s_1} n_{s_1,s_2} \cdots n_{s_{r-1},s_r} \right| & j \geq i + r \end{cases} \qquad (11.2.2)$$

The theorem now follows by taking absolute values in the expression for h_{ij} and then using (11.2.1) and (11.2.2). □

The bounds in the above theorems suggest that there is more to approximating $f(A)$ than just approximating $f(z)$ on the spectrum of A. In particular, we see that if the eigensystem of A is ill-conditioned and/or A's departure

from normality is large, then the discrepancy between $f(A)$ and $g(A)$ may be considerably larger than the maximum of $|f(z) - g(z)|$ on $\lambda(A)$. Thus, even though approximation methods avoid eigenvalue computations, they appear to be influenced by the structure of A's eigensystem, a point that we pursue further in the next section.

Example 11.2.1 Suppose

$$A = \begin{bmatrix} -.01 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & .01 \end{bmatrix}.$$

If $f(z) = e^z$ and $g(z) = 1 + z + z^2/2$, then $\| f(A) - g(A) \| \approx 10^{-5}$ in either the Frobenius norm or the 2-norm. Since $\kappa_2(X) \approx 10^7$, the error predicted by Theorem 11.2.1 is $O(1)$, rather pessimistic. On the other hand, the error predicted by the Schur decomposition approach is $O(10^{-2})$.

11.2.3 Taylor Approximants

A popular way of approximating a matrix function such as e^A is through the truncation of its Taylor series. The conditions under which a matrix function $f(A)$ has a Taylor series representation are easily established.

Theorem 11.2.3 *If $f(z)$ has a power series representation*

$$f(z) = \sum_{k=0}^{\infty} c_k z^k$$

on an open disk containing $\lambda(A)$, then

$$f(A) = \sum_{k=0}^{\infty} c_k A^k.$$

Proof. We prove the theorem for the case when A is diagonalizable. In P11.2.1, we give a hint as to how to proceed without this assumption. Suppose $X^{-1}AX = D = \text{diag}(\lambda_1, \ldots, \lambda_n)$. Using Corollary 11.1.2, we have

$$\begin{aligned} f(A) &= X\text{diag}(f(\lambda_1), \ldots, f(\lambda_n))X^{-1} \\ &= X\text{diag}\left(\sum_{k=0}^{\infty} c_k\lambda_1^k, \ldots, \sum_{k=0}^{\infty} c_k\lambda_n^k\right)X^{-1} \\ &= X\left(\sum_{k=0}^{\infty} c_k D^k\right)X^{-1} = \sum_{k=0}^{\infty} c_k(XDX^{-1})^k = \sum_{k=0}^{\infty} c_k A^k. \ \square \end{aligned}$$

Several important transcendental matrix functions have particularly simple series representations:

$$\log(I - A) = \sum_{k=1}^{\infty} \frac{A^k}{k} \qquad |\lambda| < 1,\ \lambda \in \lambda(A)$$

$$\sin(A) = \sum_{k=0}^{\infty} (-1)^k \frac{A^{2k+1}}{(2k+1)!}$$

$$\cos(A) = \sum_{k=0}^{\infty} (-1)^k \frac{A^{2k}}{(2k)!}\,.$$

The following theorem bounds the errors that arise when matrix functions such as these are approximated via truncated Taylor series.

Theorem 11.2.4 *If $f(z)$ has the Taylor series*

$$f(z) = \sum_{k=0}^{\infty} \alpha_k z^k$$

on an open disk containing the eigenvalues of $A \in \mathbb{C}^{n\times n}$, then

$$\| f(A) - \sum_{k=0}^{q} \alpha_k A^k \|_2 \le \frac{n}{(q+1)!} \max_{0\le s\le 1} \| A^{q+1} f^{(q+1)}(As) \|_2\,.$$

Proof. Define the matrix $E(s)$ by

$$f(As) = \sum_{k=0}^{q} \alpha_k (As)^k + E(s) \qquad 0 \le s \le 1\,. \tag{11.2.3}$$

If $f_{ij}(s)$ is the (i,j) entry of $f(As)$, then it is necessarily analytic and so

$$f_{ij}(s) = \left(\sum_{k=0}^{q} \frac{f_{ij}^{(k)}(0)}{k!} s^k \right) + \frac{f_{ij}^{(q+1)}(\varepsilon_{ij})}{(q+1)!} s^{q+1} \tag{11.2.4}$$

where ε_{ij} satisfies $0 \le \varepsilon_{ij} \le s \le 1$.

By comparing powers of s in (11.2.3) and (11.2.4) we conclude that $e_{ij}(s)$, the (i,j) entry of $E(s)$, has the form

$$e_{ij}(s) = \frac{f_{ij}^{(q+1)}(\varepsilon_{ij})}{(q+1)!} s^{q+1}\,.$$

Now $f_{ij}^{(q-1)}(s)$ is the (i,j) entry of $A^{q+1}f^{(q+1)}(As)$ and therefore

$$|e_{ij}(s)| \le \max_{0\le s\le 1} \frac{f_{ij}^{(q+1)}(s)}{(q+1)!} \le \max_{0\le s\le 1} \frac{\| A^{q+1}f^{(q+1)}(As) \|_2}{(q+1)!}.$$

The theorem now follows by applying (2.3.8). □

Example 11.2.2 If

$$A = \begin{bmatrix} -49 & 24 \\ -64 & 31 \end{bmatrix},$$

then

$$e^A = \begin{bmatrix} -0.735759 & .0551819 \\ -1.471518 & 1.103638 \end{bmatrix}.$$

For $q = 59$, Theorem 11.2.4 predicts that

$$\| e^A - \sum_{k=0}^{q} \frac{A^k}{k!} \|_2 \le \frac{n}{(q+1)!} \max_{0\le s\le 1} \| A^{q+1}e^{As} \|_2 \le 10^{-60}.$$

However, if $\mathbf{u} \approx 10^{-7}$, then we find

$$fl\left(\sum_{k=0}^{59} \frac{A^k}{k!}\right) = \begin{bmatrix} -22.25880 & -1.4322766 \\ -61.49931 & -3.474280 \end{bmatrix}.$$

The problem is that some of the partial sums have large elements. For example, $I + \cdots + A^{17}/17!$ has entries of order 10^7. Since the machine precision is approximately 10^{-7}, rounding errors larger than the norm of the solution are sustained.

Example 11.2.2 highlights a shortcoming of truncated Taylor series approximation: It tends to be worthwhile only near the origin. The problem can sometimes be circumvented through a change of scale. For example, by repeated application of the *double angle* formulae:

$$\cos(2A) = 2\cos(A)^2 - I \qquad \sin(2A) = 2\sin(A)\cos(A)$$

it is possible to "build up" the sine and cosine of a matrix from suitably truncated Taylor series approximates:

S_0 = Taylor approximate to $\sin(A/2^k)$
C_0 = Taylor approximate to $\cos(A/2^k)$
for $j = 1{:}k$
    $S_j = 2S_{j-1}C_{j-1}$
    $C_j = 2C_{j-1}^2 - I$
end

Here k is a positive integer chosen so that, say, $\| A \|_\infty \approx 2^k$. See Serbin and Blalock (1979).

11.2.4 Evaluating Matrix Polynomials

Since the approximation of transcendental matrix functions so often involves the evaluation of polynomials, it is worthwhile to look at the details of computing

$$p(A) = b_0 I + b_1 A + \cdots + b_q A^q$$

where the scalars $b_0, \ldots, b_q \in \mathbb{R}$ are given. The most obvious approach is to invoke Horner's scheme:

Algorithm 11.2.1 Given a matrix A and $b(0{:}q)$, the following algorithm computes $F = b_q A^q + \cdots + b_1 A + b_0 I$.

$F = b_q A + b_{q-1} I$
for $k = q-2{:}-1{:}0$
  $F = AF + b_k I$
end

This requires $q-1$ matrix multiplications. However, unlike the scalar case, this summation process is not optimal. To see why, suppose $q = 9$ and observe that

$$\begin{aligned} p(A) &= A^3(A^3(b_9A^3 + (b_8A^2 + b_7A + b_6I)) \\ &\quad +(b_5A^2 + b_4A + b_3I)) + b_2A^2 + b_1A + b_0I. \end{aligned}$$

Thus, $F = p(A)$ can be evaluated with only four matrix multiplies:

$$\begin{aligned} A_2 &= A^2 \\ A_3 &= AA_2 \\ F_1 &= b_9A_3 + b_8A_2 + b_7A + b_6I \\ F_2 &= A_3F_1 + b_5A_2 + b_4A + b_3I \\ F &= A_3F_2 + b_2A_2 + b_1A + b_0I. \end{aligned}$$

In general, if s is any integer satisfying $1 \le s \le \sqrt{q}$ then

$$p(A) = \sum_{k=0}^{r} B_k (A^s)^k \qquad r = \text{floor}(q/s) \tag{11.2.5}$$

where

$$B_k = \begin{cases} b_{sk+s-1}A^{s-1} + \cdots + b_{sk+1}A + b_{sk}I & k = 0{:}r-1 \\ b_q A^{q-sr} + \cdots + b_{sr+1}A + b_k I & k = r\,. \end{cases}$$

Once $A^2, \ldots, A^s$ are computed, Horner's rule can be applied to (11.2.5) and the net result is that $p(A)$ can be computed with $s + r - 1$ matrix

multiplies. By choosing $s = \text{floor}(\sqrt{q})$, the number of matrix multiplies is approximately minimized. This technique is discussed in Paterson and Stockmeyer (1973). Van Loan (1978) shows how the procedure can be implemented without storage arrays for $A^2, \ldots, A^s$.

11.2.5 Computing Powers of a Matrix

The problem of raising a matrix to a given power deserves special mention. Suppose it is required to compute A^{13}. Noting that $A^4 = (A^2)^2$, $A^8 = (A^4)^2$ and $A^{13} = A^8 A^4 A$, we see that this can be accomplished with just 5 matrix multiplications. In general we have

Algorithm 11.2.2 (Binary Powering) Given a positive integer s and $A \in \mathbb{R}^{n\times n}$, the following algorithm computes $F = A^s$ where s is a positive integer and $A \in \mathbb{R}^{n\times n}$.

Let $s = \sum_{k=0}^{t} \beta_k 2^k$ be the binary expansion of s with $\beta_t \neq 0$.
$Z = A$; $q = 0$
while $\beta_q = 0$
    $Z = Z^2$; $q = q + 1$
end
$F = Z$
for $k = q + 1{:}t$
    $Z = Z^2$
    if $\beta_k \neq 0$
        $F = FZ$
    end
end

This algorithm requires at most $2\,\text{floor}[\log_2(s)]$ matrix multiplies. If s is a power of 2, then only $\log_2(s)$ matrix multiplies are needed.

11.2.6 Integrating Matrix Functions

We conclude this section with some remarks on the integration of matrix functions. Suppose $f(At)$ is defined for all $t \in [a, b]$ and that we wish to compute

$$F = \int_a^b f(At)dt.$$

As in (11.1.1) the integration is on an element-by-element basis.

Ordinary quadrature rules can be applied to F. For example, with Simpson's rule, we have

$$F \approx \tilde{F} = \frac{h}{3}\sum_{k=0}^{m} w_k f(A(a+kh)) \tag{11.2.6}$$

where m is even, $h = (b-a)/m$ and

$$w_k = \begin{cases} 1 & k = 0, m \\ 4 & k \text{ odd} \\ 2 & k \text{ even}, k \neq 0, m. \end{cases}$$

If $(d^4/dz^4)f(zt) = f^{(4)}(zt)$ is continuous for $t \in [a,b]$ and if $f^{(4)}(At)$ is defined on this same interval, then it can be shown that $\tilde{F} = F + E$ where

$$\| E \|_2 \le \frac{nh^4(b-a)}{180} \max_{a \le t \le b} \| f^{(4)}(At) \|_2 . \tag{11.2.7}$$

Let f_{ij} and e_{ij} denote the (i,j) entries of F and E, respectively. Under the above assumptions we can apply the standard error bounds for Simpson's rule and obtain

$$|e_{ij}| \le \frac{h^4(b-a)}{180} \max_{a \le t \le b} |e_i^T f^{(4)}(At) e_j| .$$

The inequality (11.2.7) now follows since $\| E \|_2 \le n \max |e_{ij}|$ and

$$\max_{a \le t \le b} |e_i^T f^{(4)}(At) e_j| \le \max_{a \le t \le b} \| f^{(4)}(At) \|_2 .$$

Of course, in the practical application of (11.2.6), the function evaluations $f(A(a+kh))$ normally have to be approximated. Thus, the overall error involves the error in approximating $f(A(a+kh)$ as well as the Simpson rule error.

Problems

P11.2.1 (a) Suppose $G = \lambda I + E$ is a p-by-p Jordan block, where $E = (\delta_{i,j-1})$. Show that

$$(\lambda I + E)^k = \sum_{j=0}^{\min\{p-1,k\}} \binom{k}{j} \lambda^{k-j} E^j .$$

(b) Use (a) and Theorem 11.1.1 to prove Theorem 11.2.3.

P11.2.2 Verify (11.2.2).

P11.2.3 Show that if $\| A \|_2 < 1$, then $\log(I + A)$ exists and satisfies the bound

$\| \log(I + A) \|_2 \leq \| A \|_2/(1 - \| A \|_2)$.

P11.2.4 Let A by an n-by-n symmetric positive definite matrix. (a) Show that there exists a unique symmetric positive definite X such that $A = X^2$. (b) Show that if $X_0 = I$ and $X_{k+1} = (X_k + AX_k^{-1})/2$ then $X_k \rightarrow \sqrt{A}$ quadratically where $\sqrt{A}$ denotes the matrix X in part (a).

P11.2.5 Specialize Algorithm 11.2.1 to the case when A is symmetric. Repeat for the case when A is upper triangular. In both instances, give the associated flop counts.

P11.2.6 Show that $X(t) = C_1 \cos(t\sqrt{A}) + C_2\sqrt{A^{-1}}\sin(t\sqrt{A})$ solves the initial value problem $\ddot{X}(t) = -AX(t)$, $X(0) = C_1$, $\dot{X}(0) = C_2$. Assume that A is symmetric positive definite.

P11.2.7 Using Theorem 11.2.4, bound the error in the approximations:

$$\sin(A) \approx \sum_{k=0}^{q}(-1)^k \frac{A^{2k+1}}{(2k+1)!} \qquad \cos(A) \approx \sum_{k=0}^{q}(-1)^k \frac{A^{2k}}{(2k)!}.$$

P11.2.8 Suppose $A \in \mathbb{R}^{n\times n}$ is nonsingular and $X_0 \in \mathbb{R}^{n\times n}$ is given. The iteration defined by

$$X_{k+1} = X_k(2I - AX_k)$$

is the matrix analog of Newton's method applied to the function $f(x) = a - (1/x)$. Use the SVD to analyze this iteration. Do the iterates converge to A^{-1}? Discuss the choice of X_0.

Notes and References for Sec. 11.2

The optimality of Horner's rule for polynomial evaluation is discussed in

D. Knuth (1981). *The Art of Computer Programming , vol. 2. Seminumerical Algorithms* , 2nd ed., Addison-Wesley, Reading, Massachusetts.

M.S. Paterson and L.J. Stockmeyer (1973). "On the Number of Nonscalar Multiplications Necessary to Evaluate Polynomials," *SIAM J. Comp. 2,* 60–66.

The Horner evaluation of matrix polynomials is analyzed in

C.F. Van Loan (1978). "A Note on the Evaluation of Matrix Polynomials," *IEEE Trans. Auto. Cont. AC-24,* 320–21.

Other aspects of matrix function computation are discussed in

N.J. Higham and P.A. Knight (1995). "Matrix Powers in Finite Precision Arithmetic," *SIAM J. Matrix Anal. Appl. 16,* 343–358.

R. Mathias (1993). "Approximation of Matrix-Valued Functions," *SIAM J. Matrix Anal. Appl. 14,* 1061–1063.

S. Friedland (1991). "Revisiting Matrix Squaring," *Lin. Alg. and Its Applic. 154-156,* 59–63.

H. Bolz and W. Niethammer (1988). "On the Evaluation of Matrix Functions Given by Power Series," *SIAM J. Matrix Anal. Appl. 9,* 202–209.

The Newton and Language representations for $f(A)$ and their relationship to other matrix function definitions is discussed in

R.F. Rinehart (1955). "The Equivalence of Definitions of a Matric Function," *Amer. Math. Monthly 62,* 395–414.

The "double angle" method for computing the cosine of matrix is analyzed in

S. Serbin and S. Blalock (1979). "An Algorithm for Computing the Matrix Cosine," *SIAM J. Sci. Stat. Comp. 1*, 198–204.

The square root is a particularly important matrix function. See §4.2.10. Several approaches are possible:

Å. Björck and S. Hammarling (1983). "A Schur Method for the Square Root of a Matrix," *Lin. Alg. and Its Applic. 52/53*, 127–140.
N.J. Higham (1986). "Newton's Method for the Matrix Square Root," *Math. Comp. 46*, 537–550.
N.J. Higham (1987). "Computing Real Square Roots of a Real Matrix," *Lin. Alg. and Its Applic. 88/89*, 405–430.

11.3 The Matrix Exponential

One of the most frequently computed matrix functions is the exponential

$$e^{At} = \sum_{k=0}^{\infty} \frac{(At)^k}{k!}.$$

Numerous algorithms for computing e^{At} have been proposed, but most of them are of dubious numerical quality, as is pointed out in the survey article by Moler and Van Loan (1978). In order to illustrate what the computational difficulties are, we present a "scaling and squaring" method based upon Padé approximation. A brief analysis of the method follows that involves some e^{At} perturbation theory and comments about the shortcomings of eigenanalysis in settings where non-normality prevails.

11.3.1 A Padé Approximation Method

Following the discussion in §11.2, if $g(z) \approx e^z$, then $g(A) \approx e^A$. A very useful class of approximants for this purpose are the Padé functions defined by

$$R_{pq}(z) = D_{pq}(z)^{-1}N_{pq}(z),$$

where

$$N_{pq}(z) = \sum_{k=0}^{p} \frac{(p+q-k)!p!}{(p+q)!k!(p-k)!} z^k$$

and

$$D_{pq}(z) = \sum_{k=0}^{q} \frac{(p+q-k)!q!}{(p+q)!k!(q-k)!}(-z)^k.$$

Notice that $R_{p0}(z) = 1 + z + \cdots + z^p/p!$ is the pth order Taylor polynomial.

Unfortunately, the Padé approximants are good only near the origin, as the following identity reveals:

$$e^A = R_{pq}(A) + \frac{(-1)^q}{(p+q)!}A^{p+q+1}D_{pq}(A)^{-1}\int_0^1 u^p(1-u)^q e^{A(1-u)}du. \quad (11.3.1)$$

However, this problem can be overcome by exploiting the fact that $e^A = (e^{A/m})^m$. In particular, we can scale A by m such that $F_{pq} = R_{pq}(A/m)$ is a suitably accurate approximation to $e^{A/m}$. We then compute F_{pq}^m using Algorithm 11.2.2. If m is a power of two, then this amounts to repeated squaring and so is very efficient. The success of the overall procedure depends on the accuracy of the approximant

$$F_{pq} = \left(R_{pq}\left(\frac{A}{2^j}\right)\right)^{2^j}.$$

In Moler and Van Loan (1978) it is shown that if

$$\frac{\| A \|_\infty}{2^j} \le \frac{1}{2},$$

then there exists an $E \in \mathbb{R}^{n\times n}$ such that

$$\begin{aligned} F_{pq} &= e^{A+E} \\ AE &= EA \\ \| E \|_\infty &\le \varepsilon(p,q)\| A \|_\infty \\ \varepsilon(p,q) &= 2^{3-(p+q)}\frac{p!q!}{(p+q)!(p+q+1)!}. \end{aligned}$$

These results form the basis of an effective e^A procedure with error control. Using the above formulae it is easy to establish the inequality:

$$\frac{\| e^A - F_{pq} \|_\infty}{\| e^A \|_\infty} \le \epsilon(p,q)\| A \|_\infty e^{\epsilon(p,q)\| A \|_\infty}.$$

The parameters p and q can be determined according to some relative error tolerance. Note that since F_{pq} requires about $j + \max(p,q)$ matrix multiplies it makes sense to set $p = q$ as this choice minimizes $\epsilon(p,q)$ for a given amount of work. Encapsulating these ideas we obtain

Algorithm 11.3.1 Given $\delta > 0$ and $A \in \mathbb{R}^{n\times n}$, the following algorithm computes $F = e^{A+E}$ where $\| E \|_\infty \le \delta\| A \|_\infty$

$j = \max(0, 1 + \text{floor}(\log_2(\| A \|_\infty)))$
$A = A/2^j$
Let q be the smallest non-negative integer such that $\epsilon(q,q) \le \delta$.

$D = I$; $N = I$; $X = I$; $c = 1$
for $k = 1{:}q$
    $c = c(q-k+1)/[(2q-k+1)k]$
    $X = AX$; $N = N + cX$; $D = D + (-1)^k cX$
end
Solve $DF = N$ for F using Gaussian elimination.
for $k = 1{:}j$
    $F = F^2$
end

This algorithm requires about $2(q + j + 1/3)n^3$flops. The roundoff error properties of have essentially been analyzed by Ward (1977).

The special Horner techniques of §11.2 can be applied to quicken the computation of $D = D_{qq}(A)$ and $N = N_{qq}(A)$. For example, if $q = 8$ we have $N_{qq}(A) = U + AV$ and $D_{qq}(A) = U - AV$ where

$$U = c_0 I + c_2 A^2 + (c_4 I + c_6 A^2 + c_8 A^4)A^4$$

and

$$V = c_1 I + c_3 A^2 + (c_5 I + c_7 A^2)A^4.$$

Clearly, N and D can be found in 5 matrix multiplies rather than the 7 required by Algorithm 11.3.1.

11.3.2 Perturbation Theory

Is Algorithm 11.3.1 stable in the presence of roundoff error? To answer this question we need to understand the sensitivity of the matrix exponential to perturbations in A. The starting point in the discussion is the initial value problem

$$\dot{X}(t) = AX(t) \qquad X(0) = I$$

where $A, X(t) \in \mathbb{R}^{n\times n}$. This has the unique solution $X(t) = e^{At}$, a characterization of the matrix exponential that can be used to establish the identity

$$e^{(A+E)t} - e^{At} = \int_0^t e^{A(t-s)} E e^{(A+E)s} ds\,.$$

From this it follows that

$$\frac{\| e^{(A+E)t} - e^{At} \|_2}{\| e^{At} \|_2} \le \frac{\| E \|_2}{\| e^{At} \|_2} \int_0^t \| e^{A(t-s)} \|_2 \, \| e^{(A+E)s} \|_2 ds\,.$$

Further simplifications result if we bound the norms of the exponentials that appear in the integrand. One way of doing this is through the Schur decomposition. If $Q^H AQ = \text{diag}(\lambda_i) + N$ is the Schur decomposition of $A \in \mathbb{C}^{n\times n}$, then it can be shown that

$$\| e^{At} \|_2 \le e^{\alpha(A)t} M_S(t), \tag{11.3.2}$$

where

$$\alpha(A) = \max\{\mathrm{Re}(\lambda) : \lambda \in \lambda(A)\} \tag{11.3.3}$$

and

$$M_S(t) = \sum_{k=0}^{n-1} \frac{\| Nt \|_2^k}{k!}.$$

The quantity $\alpha(A)$ is called the *spectral abscissa* and with a little manipulation it can be shown that

$$\frac{\| e^{(A+E)t} - e^{At} \|_2}{\| e^{At} \|_2} \le t\| E \|_2 M_S(t)^2 \exp(tM_S(t)\| E \|_2).$$

Notice that $M_S(t) \equiv 1$ if and only if A is normal, suggesting that the matrix exponential problem is "well behaved" if A is normal. This observation is confirmed by the behavior of the *matrix exponential condition number* $\nu(A,t)$, defined by

$$\nu(A,t) = \max_{\| E \|_2 \le 1} \left\| \int_0^t e^{A(t-s)} E e^{As} ds \right\|_2 \frac{\| A \|_2}{\| e^{At} \|_2}.$$

This quantity, discussed in Van Loan (1977), measures the sensitivity of the map $A \to e^{At}$ in that for a given t, there is a matrix E for which

$$\frac{\| e^{(A+E)t} - e^{At} \|_2}{\| e^{At} \|_2} \approx \nu(A,t)\frac{\| E \|_2}{\| A \|_2}.$$

Thus, if $\nu(A,t)$ is large, small changes in A can induce relatively large changes in e^{At}. Unfortunately, it is difficult to characterize precisely those A for which $\nu(A,t)$ is large. (This is in contrast to the linear equation problem $Ax = b$, where the ill-conditioned A are neatly described in terms of SVD.) One thing we can say, however, is that $\nu(A,t) \ge t\| A \|_2$, with equality holding for all non-negative t if and only if A is normal.

Dwelling a little more on the effect of non-normality, we know from the analysis of §11.2 that approximating e^{At} involves more than just approximating e^{zt} on $\lambda(A)$. Another clue that eigenvalues do not "tell the whole story" in the e^{At} problem has to do with the inability of the spectral abscissa (11.3.3) to predict the size of $\| e^{At} \|_2$ as a function of time. If A is normal, then

$$\| e^{At} \|_2 = e^{\alpha(A)t} \tag{11.3.4}$$

Thus, there is uniform decay if the eigenvalues of A are in the open left half plane. But if A is non-normal, then e^{At} can grow before decay "sets in." The 2-by-2 example

$$A = \begin{bmatrix} -1 & M \\ 0 & -1 \end{bmatrix} \Leftrightarrow \qquad e^{At} = e^{-t}\begin{bmatrix} 1 & tM \\ 0 & 1 \end{bmatrix}$$

plainly illustrates this point.

11.3.3 Some Stability Issues

With this discussion we are ready to begin thinking about the stability of Algorithm 11.3.1. A potential difficulty arises during the squaring process if A is a matrix whose exponential grows before it decays. If

$$G = R_{qq}\left(\frac{A}{2^j}\right) \approx e^{A/2^j},$$

then it can be shown that rounding errors of order

$$\gamma = \mathbf{u}\| G^2 \|_2 \| G^4 \|_2 \| G^8 \|_2 \cdots \| G^{2^{j-1}} \|_2$$

can be expected to contaminate the computed G^{2^j}. If $\| e^{At} \|_2$ has a substantial initial growth, then it may be the case that

$$\gamma \gg \mathbf{u}\| G^{2^j} \|_2 \approx \mathbf{u}\| e^A \|_2$$

thus ruling out the possibility of small relative errors.

If A is normal, then so is the matrix G and therefore $\| G^m \|_2 = \| G \|_2^m$ for all positive integers m. Thus, $\gamma \approx \mathbf{u}\| G^{2^j} \|_2 \approx \mathbf{u}\| e^A \|_2$ and so the initial growth problems disappear. The algorithm can essentially be guaranteed to produce small relative error when A is normal. On the other hand, it is more difficult to draw conclusions about the method when A is non-normal because the connection between $\nu(A, t)$ and the initial growth phenomena is unclear. However, numerical experiments suggest that Algorithm 11.3.1 fails to produce a relatively accurate e^A only when $v(A, 1)$ is correspondingly large.

11.3.4 Eigenvalues and Pseudo-Eigenvalues

We closed §7.1 with a comment that the eigenvalues of a matrix are generally not good "informers" when it comes to measuring nearness to singularity, unless the matrix is normal. It is the singular values that shed light on $Ax = b$ sensitivity. Our discussion of the matrix exponential is another warning to the same effect. The spectrum of a non-normal A does not completely describe e^{At} behavior.

In many applications, the eigenvalues of a matrix "say something" about an underlying phenomenon that is being modeled. If the eigenvalues are extremely sensitive to perturbation, then what they say can be misleading. This has prompted the development of the idea of pseudospectra. For $\epsilon \geq 0$, the *ϵ-pseudospectrum* of a matrix A is a subset of the complex plane defined by

$$\lambda_\epsilon(A) = \left\{ z \in \mathbb{C} : \| (zI - A)^{-1} \|_2 \geq \frac{1}{\epsilon} \right\} \tag{11.3.5}$$

Qualitatively, z is a pseudo-eigenvalue of A if $zI - A$ is sufficiently close to singular. By convention we set $\lambda_0(A) = \lambda(A)$. Here are some pseudospectra properties:

1. If $\epsilon_1 \leq \epsilon_2$, then $\lambda_{\epsilon_1}(A) \subseteq \lambda_{\epsilon_2}(A)$.
2. $\lambda_\epsilon(A) = \{z \in \mathbb{C} : \sigma_{min}(zI - A) \leq \epsilon\}$.
3. $\lambda_\epsilon(A) = \{z \in \mathbb{C} : z \in \lambda(A+E), \text{for some } E \text{ with } \| E \|_2 \leq \epsilon\}$.

Plotting the pseudospectra of a non-normal matrix A can provide insight into behavior. Here "behavior" can mean anything from the mathematical behavior of an iteration to solve $Ax = b$ to the physical behavior predicted by a model that involves A. See Higham and Trefethen (1993), Nachtigal, Reddy, and Trefethen (1992), and Trefethen, Trefethen, Reddy, and Driscoll (1993).

Problems

P11.3.1 Show that $e^{(A+B)t} = e^{At}e^{Bt}$ for all t if and only if $AB = BA$. (Hint: Express both sides as a power series in t and compare the coefficient of t.)

P11.3.2 Suppose that A is skew-symmetric. Show that both e^A and the (1,1) Padé approximate $R_{11}(A)$ are orthogonal. Are there any other values of p and q for which $R_{pq}(A)$ is orthogonal?

P11.3.3 Show that if A is nonsingular, then there exists a matrix X such that $A = e^X$. Is X unique?

P11.3.4 Show that if

$$\exp\left(\begin{bmatrix} -A^T & P \\ 0 & A \end{bmatrix} z\right) = \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix} \begin{matrix} n \\ n \end{matrix}$$
$$\qquad\qquad\qquad\qquad\qquad\quad n \quad\;\; n$$

then

$$F_{11}^T F_{12} = \int_0^z e^{A^T t} P e^{At} dt.$$

P11.3.5 Give an algorithm for computing e^A when $A = uv^T$, $u, v \in \mathbb{R}^n$.

P11.3.6 Suppose $A \in \mathbb{R}^{n \times n}$ and that $v \in \mathbb{R}^n$ has unit 2-norm. Define the function $\phi(t) = \| e^{At}v \|_2^2/2$ and show that

$$\dot{\phi}(t) \leq \mu(A)\phi(t)$$

where $\mu(A) = \lambda_1((A + A^T)/2)$. Conclude that $\| e^{At} \|_2 \leq e^{\mu(A)t}$ where $t \geq 0$.

P11.3.7 Prove the three pseudospectra properties given in the text.

Notes and References for Sec. 11.3

Much of what appears in this section and an extensive bibliography may be found in the following survey article:

C.B. Moler and C.F. Van Loan (1978). "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review 20*, 801–36.

Scaling and squaring with Padé approximants (Algorithm 11.3.1) and a careful implementation of Parlett's Schur decomposition method (Algorithm 11.1.1) were found to be among the less dubious of the nineteen methods scrutinized. Various aspects of Padé

approximation of the matrix exponential are discussed in

W. Fair and Y. Luke (1970). "Padé Approximations to the Operator Exponential," *Numer. Math. 14*, 379–82.
C.F. Van Loan (1977). "On the Limitation and Application of Padé Approximation to the Matrix Exponential," in *Padé and Rational Approximation*, ed. E.B. Saff and R.S. Varga, Academic Press, New York.
R.C. Ward (1977). "Numerical Computation of the Matrix Exponential with Accuracy Estimate," *SIAM J. Num. Anal. 14*, 600–14.
A. Wragg (1973). "Computation of the Exponential of a Matrix I: Theoretical Considerations," *J. Inst. Math. Applic. 11*, 369–75.
A. Wragg (1975). "Computation of the Exponential of a Matrix II: Practical Considerations," *J. Inst. Math. Applic. 15*, 273–78.

A proof of equation (11.3.1) for the scalar case appears in

R.S. Varga (1961). "On Higher-Order Stable Implicit Methods for Solving Parabolic Partial Differential Equations," *J. Math. Phys. 40*, 220–31.

There are many applications in control theory calling for the computation of the matrix exponential. In the linear optimal regular problem, for example, various integrals involving the matrix exponential are required. See

J. Johnson and C.L. Phillips (1971). "An Algorithm for the Computation of the Integral of the State Transition Matrix," *IEEE Trans. Auto. Cont. AC-16*, 204–5.
C.F. Van Loan (1978). "Computing Integrals Involving the Matrix Exponential," *IEEE Trans. Auto. Cont. AC-23*, 395–404.

An understanding of the map $A \rightarrow \exp(At)$ and its sensitivity is helpful when assessing the performance of algorithms for computing the matrix exponential. Work in this direction includes

B. Kågström (1977). "Bounds and Perturbation Bounds for the Matrix Exponential," *BIT 17*, 39–57.
C.F. Van Loan (1977). "The Sensitivity of the Matrix Exponential," *SIAM J. Num. Anal. 14*, 971–81.
R. Mathias (1992). "Evaluating the Frechet Derivative of the Matrix Exponential," *Numer. Math. 63*, 213–226.

The computation of a logarithm of a matrix is an important area demanding much more work. These calculations arise in various "system identification" problems. See

B. Singer and S. Spilerman (1976). "The Representation of Social Processes by Markov Models," *Amer. J. Sociology 82*, 1–54.
B.W. Helton (1968). "Logarithms of Matrices," *Proc. Amer. Math. Soc. 19*, 733 36.

For pointers into the pseudospectra literature we recommend

L.N. Trefethen (1992). "Pseudospecta of Matrices," in *Numerical Analysis 1991*, D.F. Griffiths and G.A. Watson (eds), Longman Scientific and Technical, Harlow, Essex, UK, 234-262.
D.J. Higham and L.N. Trefethen (1993). "Stiffness of ODES," *BIT 33*, 285–303.
L.N. Trefethen, A.E. Trefethen, S.C. Reddy, and T.A. Driscoll (1993). "Hydrodynamic Stability Without Eigenvalues," *Science 261*, 578-584.

as well as Chaitin-Chatelin and Frayssé (1996, chapter 10).

Chapter 12

Special Topics

In this final chapter we discuss an assortment of problems that represent important applications of the singular value, QR, and Schur decompositions. We first consider least squares minimization with constraints. Two types of constraints are considered in §12.1, quadratic inequality and linear equality. The next two sections are also concerned with variations on the standard LS problem. In §12.2 we consider how the vector of observations b might be approximated by some subset of A's columns, a course of action that is sometimes appropriate if A is rank-deficient. In §12.3 we consider a variation of ordinary regression known as total least squares that has appeal when A is contaminated with error. More applications of the SVD are considered in §12.4, where various subspace calculations are considered. In §12.5 we investigate the updating of orthogonal factorizations when the matrix A undergoes a low-rank perturbation. Some variations of the basic eigenvalue problem are discussed in §12.6.

Before You Begin

Because of the topical nature of this chapter, it doesn't make sense to have a chapter-wide, before-you-begin advisory. Instead, each section will begin with pointers to earlier portions of the book, and, if appropriate, pointers to LAPACK and other texts.

12.1 Constrained Least Squares

In the least squares setting it is sometimes natural to minimize $\| Ax - b \|_2$ over a proper subset of $\mathbb{R}^n$. For example, we may wish to predict b as best we can with Ax subject to the constraint that x is a unit vector. Or, perhaps the solution defines a fitting function $f(t)$ which is to have prescribed values at a finite number of points. This can lead to an equality constrained least squares problem. In this section we show how these problems can be solved using the QR factorization and the SVD.

Chapter 5 and §8.7 should be understood before reading this section. LAPACK connections include:

LAPACK: Tools for Generalized/Constrained LS Problems	
_GGLSE	Solves the equality constrained LS problem
_GGQRF	Computes the generalized QR factorization of a matrix pair
_GGRQF	Computes the generalized RQ factorization of a matrix pair
_GGSVP	Converts the GSVD problem to triangular form
_TGSJA	Computes the GSVD of a pair of triangular matrices

Complementary references include Lawson and Hanson (1974) and Björck (1996).

12.1.1 The Problem LSQI

Least squares minimization with a quadratic inequality constraint—the *LSQI problem*—is a technique that can be used whenever the solution to the ordinary LS problem needs to be *regularized*. A simple LSQI problem that arises when attempting to fit a function to noisy data is

$$\text{minimize } \| Ax - b \|_2 \qquad \text{subject to } \| Bx \|_2 \leq \alpha \tag{12.1.1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $B \in \mathbb{R}^{n\times n}$ (nonsingular), and $\alpha \geq 0$. The constraint defines a hyperellipsoid in $\mathbb{R}^n$ and is usually chosen to damp out excessive oscillation in the fitting function. This can be done, for example, if B is a discretized second derivative operator.

More generally, we have the problem

$$\text{minimize } \| Ax - b \|_2 \qquad \text{subject to } \| Bx - d \|_2 \leq \alpha \tag{12.1.2}$$

where $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$, $b \in \mathbb{R}^m$, $B \in \mathbb{R}^{p\times n}$, $d \in \mathbb{R}^p$, and $\alpha \geq 0$. The generalized singular value decomposition of §8.7.3 sheds light on the solvability of (12.1.2). Indeed, if

$$\begin{aligned} U^T AX &= \text{diag}(\alpha_1, \ldots, \alpha_n) \qquad U^T U = I_m \\ V^T BX &= \text{diag}(\beta_1, \ldots, \beta_q) \qquad V^T V = I_p, \quad q = \min\{p, n\} \end{aligned} \tag{12.1.3}$$

is the generalized singular value decomposition of A and B, then (12.1.2) transforms to

$$\text{minimize } \| D_A y - \tilde{b} \|_2 \qquad \text{subject to } \| D_B y - \tilde{d} \|_2 \le \alpha$$

where $\tilde{b} = U^T b$, $\tilde{d} = V^T d$, and $y = X^{-1}x$. The simple form of the objective function

$$\| D_A y - \tilde{b} \|_2^2 \;=\; \sum_{i=1}^{n} (\alpha_i y_i - \tilde{b}_i)^2 \;+\; \sum_{i=n+1}^{m} \tilde{b}_i^2 \qquad (12.1.4)$$

and the constraint equation

$$\| D_B y - \tilde{d} \|_2^2 \;=\; \sum_{i=1}^{r} (\beta_i y_i - \tilde{d}_i)^2 \;+\; \sum_{i=r+1}^{p} \tilde{d}_i^2 \;\le\; \alpha^2 \qquad (12.1.5)$$

facilitate the analysis of the LSQI problem. Here, r = rank(B) and we assume that $\beta_{r+1} = \cdots = \beta_q = 0$.

To begin with, the problem has a solution if and only if

$$\sum_{i=r+1}^{p} \tilde{d}_i^2 \;\le\; \alpha^2 .$$

If we have equality in this expression then consideration of (12.1.4) and (12.1.5) shows that the vector defined by

$$y_i = \begin{cases} \tilde{d}_i/\beta_i & i = 1{:}r \\ \tilde{b}_i/\alpha_i & i = r+1{:}n, \alpha_i \ne 0 \\ 0 & i = r+1{:}n, \alpha_i = 0 \end{cases} \qquad (12.1.6)$$

solves the LSQI problem. Otherwise

$$\sum_{i=r+1}^{p} \tilde{d}_i^2 \;<\; \alpha^2. \qquad (12.1.7)$$

and we have more alternatives to pursue. The vector $y \in \mathbb{R}^n$, defined by

$$y_i \;=\; \begin{cases} \tilde{b}_i/\alpha_i & \alpha_i \ne 0 \\ \tilde{d}_i/\beta_i & \alpha_i = 0 \end{cases} \qquad i = 1{:}n$$

is a minimizer of $\| D_A y - \tilde{b} \|_2$. If this vector is also feasible, then we have a solution to (12.1.2). (This is not necessarily the solution of minimum 2-norm, however.) We therefore assume that

$$\sum_{\substack{i=1 \\ \alpha_i \ne 0}}^{q} \left(\beta_i \frac{\tilde{b}_i}{\alpha_i} - \tilde{d}_i \right)^2 \;+\; \sum_{i=q+1}^{p} \tilde{d}_i^2 \;>\; \alpha^2. \qquad (12.1.8)$$

This implies that the solution to the LSQI problem occurs on the boundary of the feasible set. Thus, our remaining goal is to

$$\text{minimize } \| D_A y - \tilde{b} \|_2 \qquad \text{subject to } \| D_B y - \tilde{d} \|_2 = \alpha .$$

To solve this problem, we use the method of Lagrange multipliers. Defining

$$h(\lambda, y) = \| D_A y - \tilde{b} \|_2^2 \; + \lambda \left(\| D_B y - \tilde{d} \|_2^2 \; - \alpha^2 \right)$$

we see that the equations $0 = \partial h / \partial y_i$, $i = 1{:}n$, lead to the linear system

$$(D_A^T D_A + \lambda D_B^T D_B) y = D_A^T \tilde{b} + \lambda D_B^T \tilde{d}.$$

Assuming that the matrix of coefficients is nonsingular, this has a solution $y(\lambda)$ where

$$y_i(\lambda) \;=\; \begin{cases} \dfrac{\alpha_i \tilde{b}_i + \lambda \beta_i \tilde{d}_i}{\alpha_i^2 + \lambda \beta_i^2} & i = 1{:}q \\ \tilde{b}_i / \alpha_i & i = q+1{:}n \end{cases}$$

To determine the Lagrange parameter we define,

$$\phi(\lambda) \;\equiv\; \| D_B y(\lambda) - \tilde{d} \|_2^2 \;=\; \sum_{i=1}^{r} \left(\alpha_i \frac{\beta_i \tilde{b}_i - \alpha_i \tilde{d}_i}{\alpha_i^2 + \lambda \beta_i^2} \right)^2 \;+\; \sum_{i=r+1}^{p} \tilde{d}_i^2$$

and seek a solution to $\phi(\lambda) = \alpha^2$. Equations of this type are referred to as *secular equations* and we encountered them earlier in §8.5.3. From (12.1.8) we see that $\phi(0) > \alpha^2$. Now $\phi(\lambda)$ is monotone decreasing for $\lambda > 0$, and (12.1.8) therefore implies the existence of a unique positive λ^* for which $\phi(\lambda^*) = \alpha^2$. It is easy to show that this is the desired root. It can be found through the application of any standard root-finding technique, such as Newton's method. The solution of the original LSQI problem is then $x = Xy(\lambda^*)$.

12.1.2 LS Minimization Over a Sphere

For the important case of minimization over a sphere ($B = I_n$, $d = 0$), we have the following procedure:

Algorithm 12.1.1 Given $A \in \mathbb{R}^{m \times n}$ with $m \geq n$, $b \in \mathbb{R}^m$, and $\alpha > 0$, the following algorithm computes a vector $x \in \mathbb{R}^n$ such that $\| Ax - b \|_2$ is minimum, subject to the constraint that $\| x \|_2 \leq \alpha$.

Compute the SVD $A = U \Sigma V^T$,save $V = [\, v_1, \ldots, v_n \,]$, and
    form $b = U^T b$.
$r = \text{rank}(A)$

if $\sum_{i=1}^{r}\left(\frac{b_i}{\sigma_i}\right)^2 > \alpha^2$

Find λ^* such that $\sum_{i=1}^{r}\left(\frac{\sigma_i b_i}{\sigma_i^2+\lambda^*}\right)^2 = \alpha^2$.

$x = \sum_{i=1}^{r}\left(\frac{\sigma_i b_i}{\sigma_i^2+\lambda^*}\right) v_i$

else

$x = \sum_{i=1}^{r}\left(\frac{b_i}{\sigma_i}\right) v_i$

end

The SVD is the dominant computation in this algorithm.

Example 12.1.1 The secular equation for the problem

$$\min_{\| x \|_2 = 1} \left\| \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 4 \\ 2 \\ 3 \end{bmatrix} \right\|_2$$

is given by

$$\left(\frac{8}{\lambda+4}\right)^2 + \left(\frac{2}{\lambda+1}\right)^2 = 1 .$$

For this problem we find $\lambda^* = 4.57132$ and $x = [.93334\ .35898]^T$.

12.1.3 Ridge Regression

The problem solved by Algorithm 12.1.1 is equivalent to the Lagrange multiplier problem of determining $\lambda > 0$ such that

$$(A^TA + \lambda I)x = A^Tb \qquad (12.1.9)$$

and $\| x \|_2 = \alpha$. This equation is precisely the normal equation formulation for the *ridge regression* problem

$$\min_x \left\| \begin{bmatrix} A \\ \sqrt{\lambda} I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2 = \min_x \| Ax - b \|_2^2 + \lambda \| x \|_2^2 .$$

In the general ridge regression problem one has some criteria for selecting the ridge parameter λ, e.g., $\| x(\lambda) \|_2 = \alpha$ for some given α. We describe a λ-selection procedure that is discussed in Golub, Heath, and Wahba (1979).

Set $D_k = I - e_k e_k^T = \text{diag}(1, \ldots, 1, 0, 1, \ldots, 1) \in \mathbb{R}^{m \times m}$ and let $x_k(\lambda)$ solve

$$\min_x \| D_k(Ax - b) \|_2^2 + \lambda \| x \|_2^2 . \qquad (12.1.10)$$

Thus, $x_k(\lambda)$ is the solution to the ridge regression problem with the kth row of A and kth component of b deleted, i.e., the kth experiment is ignored. Now consider choosing λ so as to minimize the *cross-validation weighted square error* $C(\lambda)$ defined by

$$C(\lambda) \;=\; \frac{1}{m}\sum_{k=1}^{m} w_k(a_k^T x_k(\lambda) - b_k)^2 \, .$$

Here, $w_1, \ldots, w_m$ are non-negative weights and a_k^T is the kth row of A. Noting that

$$\| Ax_k(\lambda) - b \|_2^2 \;=\; \| D_k(Ax_k(\lambda) - b) \|_2^2 \;+ (a_k^T x_k(\lambda) - b_k)^2$$

we see that $\left(a_k^T x_k(\lambda) - b_k\right)^2$ is the increase in the sum of squares resulting when the kth row is "reinstated." Minimizing $C(\lambda)$ is tantamount to choosing λ such that the final model is not overly dependent on any one experiment.

A more rigorous analysis can make this statement precise and also suggest a method for minimizing $C(\lambda)$. Assuming that $\lambda > 0$, an algebraic manipulation shows that

$$x_k(\lambda) \;=\; x(\lambda) \;+\; \frac{a_k^T x(\lambda) - b_k}{1 - z_k^T a_k} z_k \qquad (12.1.11)$$

where $z_k \;=\; (A^TA + \lambda I)^{-1} a_k$ and $x(\lambda) \;=\; (A^TA + \lambda I)^{-1}A^Tb$. Applying $-a_k^T$ to (12.1.11) and then adding b_k to each side of the resulting equation gives

$$b_k - a_k^T x_k(\lambda) \;=\; \frac{e_k^T(I - A(A^TA + \lambda I)^{-1}A^T)b}{e_k^T(I - A(A^TA + \lambda I)^{-1}A^T)e_k} \, . \qquad (12.1.12)$$

Noting that the residual $r \;=\; (r_1, \ldots, r_m)^T \;=\; b \,-\, Ax(\lambda)$ is given by the formula $r = [I - A(A^TA + \lambda I)^{-1}A^T]b$, we see that

$$C(\lambda) \;=\; \frac{1}{m}\sum_{k=1}^{m} w_k \left(\frac{r_k}{\partial r_k / \partial b_k}\right)^2 \, .$$

The quotient $r_k/(\partial r_k/\partial b_k)$ may be regarded as an inverse measure of the "impact" of the kth observation b_k on the model. When $\partial r_k/\partial b_k$ is small, this says that the error in the model's prediction of b_k is somewhat independent of b_k. The tendency for this to be true is lessened by basing the model on the λ^* that minimizes $C(\lambda)$.

The actual determination of λ^* is simplified by computing the SVD of A. Indeed, if $U^TAV \;=\; \text{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \ldots \geq \sigma_n$ and $\tilde{b} = U^Tb$,

then it can be shown from (12.1.12) that

$$C(\lambda) = \frac{1}{m}\sum_{k=1}^{m} w_k \left[\frac{\tilde{b}_k - \sum_{j=1}^{r} u_{kj}\tilde{b}_j \left(\frac{\sigma_j^2}{\sigma_j^2 + \lambda} \right)}{1 - \sum_{j=1}^{r} u_{kj}^2 \left(\frac{\sigma_j^2}{\sigma_j^2 + \lambda} \right)} \right]^2 .$$

The minimization of this expression is discussed in Golub, Heath, and Wahba (1979).

12.1.4 Equality Constrained Least Squares

We conclude the section by considering the least squares problem with linear equality constraints:

$$\min_{Bx=d} \| Ax - b \|_2 \tag{12.1.13}$$

Here $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times n}$, $b \in \mathbb{R}^m$, $d \in \mathbb{R}^p$, and $\text{rank}(B) = p$. We refer to (12.1.13) as the *LSE* problem. By setting $\alpha = 0$ in (12.1.2) we see that the LSE problem is a special case of the LSQI problem. However, it is simpler to approach the LSE problem directly rather than through Lagrange multipliers.

Assume for clarity that both A and B have full rank. Let

$$Q^T B^T = \begin{bmatrix} R \\ 0 \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix}$$

be the QR factorization of B^T and set

$$AQ = \underset{\quad p \qquad n-p}{[\; A_1 \quad A_2 \;]} \qquad Q^T x = \begin{bmatrix} y \\ z \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix} .$$

It is clear that with these transformations (12.1.13) becomes

$$\min_{R^T y = d} \| A_1 y + A_2 z - b \|_2.$$

Thus, y is determined from the constraint equation $R^T y = d$ and the vector z is obtained by solving the unconstrained *LS* problem

$$\min_z \| A_2 z - (b - A_1 y) \|_2.$$

Combining the above, we see that $x = Q \begin{bmatrix} y \\ z \end{bmatrix}$ solves (12.1.13).

Algorithm 12.1.2 Suppose $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^m$, and $d \in \mathbb{R}^p$. If $\text{rank}(A) = n$ and $\text{rank}(B) = p$, then the following algorithm minimizes $\| Ax - b \|_2$ subject to the constraint $Bx = d$.

$B^T = QR$ (QR factorization)
Solve $R(1{:}p, 1{:}p)^T y = d$ for y.
$A = AQ$
Find z so $\| A(:, p+1{:}n)z - (b - A(:, 1{:}p)y) \|_2$ is minimized.
$x = Q(:, 1{:}p)y + Q(:, p+1{:}n)z$

Note that this approach to the LSE problem involves two factorizations and a matrix multiplication.

12.1.5 The Method of Weighting

An interesting way to obtain an approximate solution to (12.1.13) is to solve the unconstrained LS problem

$$\min_x \left\| \begin{bmatrix} A \\ \lambda B \end{bmatrix} x - \begin{bmatrix} b \\ \lambda d \end{bmatrix} \right\|_2 \tag{12.1.14}$$

for large λ. The generalized singular value decomposition of §8.7.3 sheds light on the quality of the approximation. Let

$$\begin{aligned} U^T A X &= \text{diag}(\alpha_1, \ldots, \alpha_n) = D_A \in \mathbb{R}^{m \times n} \\ V^T B X &= \text{diag}(\beta_1, \ldots, \beta_p) = D_B \in \mathbb{R}^{p \times n} \end{aligned}$$

be the GSVD of (A, B) and assume that both matrices have full rank for clarity. If $U = [\, u_1, \ldots, u_m \,]$, $V = [\, v_1, \ldots, v_p \,]$, and $X = [\, x_1, \ldots, x_n \,]$, then it is easy to show that

$$x = \sum_{i=1}^{p} \frac{v_i^T d}{\beta_i} x_i + \sum_{i=p+1}^{n} \frac{u_i^T b}{\alpha_i} x_i \tag{12.1.15}$$

is the exact solution to (12.1.13), while

$$x(\lambda) = \sum_{i=1}^{p} \frac{\alpha_i u_i^T b + \lambda^2 \beta_i^2 v_i^T d}{\alpha_i^2 + \lambda^2 \beta_i^2} x_i + \sum_{i=p+1}^{n} \frac{u_i^T b}{\alpha_i} x_i \tag{12.1.16}$$

solves (12.1.14). Since

$$x(\lambda) - x = \sum_{i=1}^{p} \frac{\alpha_i (\beta_i u_i^T b - \alpha_i v_i^T d)}{\beta_i (\alpha_i^2 + \lambda^2 \beta_i^2)} x_i \tag{12.1.17}$$

it follows that $x(\lambda) \to x$ as $\lambda \to \infty$.

The appeal of this approach to the LSE problem is that no special subroutines are required: an ordinary LS solver will do. However, for large values of λ numerical problems can arise and it is necessary to take precautions. See Powell and Reid (1968) and Van Loan (1982a).

Example 12.1.2 The problem

$$\min_{x_1=x_2} \left\| \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 7 \\ 1 \\ 3 \end{bmatrix} \right\|_2$$

has solution $x = [.3407821\ ,\ .3407821]^T$. This can be approximated by solving

$$\min \left\| \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 1000 & -1000 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 7 \\ 1 \\ 3 \\ 0 \end{bmatrix} \right\|_2$$

which has solution $x = [.3407810\ ,\ .3407829]^T$.

Problems

P 12.1.1 (a) Show that if $\text{null}(A) \cap \text{null}(B) \neq \{0\}$, then (12.1.2) cannot have a unique solution. (b) Give an example which shows that the converse is not true. (Hint: A^+b feasible.)

P12.1.2 Let $p_0(x), \ldots, p_n(x)$ be given polynomials and $(x_0, y_0), \ldots, (x_m, y_m)$ a given set of coordinate pairs with $x_i \in [a, b]$. It is desired to find a polynomial $p(x) = \sum_{k=0}^{n} a_k p_k(x)$ such that $\sum_{i=0}^{m} (p(x_i) - y_i)^2$ is minimized subject to the constraint that

$$\int_a^b [p''(x)]^2 dx \approx h \sum_{i=0}^{N} \left(\frac{p(z_{i-1}) - 2p(z_i) + p(z_{i+1})}{h^2} \right)^2 \leq \alpha^2$$

where $z_i = a + ih$ and $b = a + Nh$. Show that this leads to an LSQI problem of the form (12.1.1).

P12.1.3 Suppose $Y = [\, y_1, \ldots, y_k \,] \in \mathbb{R}^{m \times k}$ has the property that

$$Y^T Y = \text{diag}(d_1^2, \ldots, d_k^2) \qquad d_1 \geq d_2 \geq \cdots \geq d_k > 0\,.$$

Show that if $Y = QR$ is the QR factorization of Y, then R is diagonal with $|r_{ii}| = d_i$.

P12.1.4 (a) Show that if $(A^T A + \lambda I)x = A^T b$, $\lambda > 0$, and $\| x \|_2 = \alpha$, then $z = (Ax - b)/\lambda$ solves the *dual equations* $(AA^T + \lambda I)z = -b$ with $\| A^T z \|_2 = \alpha$. (b) Show that if $(AA^T + \lambda I)z = -b$, $\| A^T z \|_2 = \alpha$, then $x = -A^T z$ satisfies $(A^T A + \lambda I)x = A^T b$, $\| x \|_2 = \alpha$.

P12.1.5 Suppose A is the m-by-1 matrix of ones and let $b \in \mathbb{R}^m$. Show that the cross-validation technique with unit weights prescribes an optimal λ given by

$$\lambda = \left(\left(\frac{\bar{b}}{s} \right)^2 - \frac{1}{m} \right)^{-1}$$

where $\bar{b}^T = (b_1 + \cdots + b_m)/m$ and $s = \sum_{i=1}^{m} (b_i - \bar{b})^2/(m-1)$.

P12.1.6 Establish equations (12.1.15), (12.1.16), and (12.1.17).

P12.1.7 Develop an SVD version of Algorithm 12.1.2 that can handle rank deficiency in A and B.

P12.1.8 Suppose

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

where $A_1 \in \mathbb{R}^{n\times n}$ is nonsingular and $A_2 \in \mathbb{R}^{(m-n)\times n}$. Show that

$$\sigma_{min}(A) \;\geq\; \sqrt{1 + \sigma_{min}(A_2A_1^{-1})^2}\,\sigma_{min}(A_1)\,.$$

P12.1.9 Consider the problem

$$\min_{\substack{x^TBx=\beta^2 \\ x^TCx=\gamma^2}} \| Ax - b \|_2 \qquad A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ B, C \in \mathbb{R}^{n\times n}$$

Assume that B and C are positive definite and that $Z \in \mathbb{R}^{n\times n}$ is a nonsingular matrix with the property that $Z^TBZ = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $Z^TCZ = I_n$. Assume that $\lambda_1 \geq \cdots \geq \lambda_n$. (a) Show that the the set of feasible x is empty unless $\lambda_n \leq \beta^2/\gamma^2 \leq \lambda_1$. (b) Using Z, show how the two constraint problem can be converted to a single constraint problem of the form

$$\min_{y^TWy=\beta^2-\lambda_n\gamma^2} \| \tilde{A}x - b \|_2$$

where $W = \text{diag}(\lambda_1, \ldots, \lambda_n) - \lambda_n I$.

P12.1.10 Suppose $p \geq m \geq n$ and that $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times p}$ Show how to compute orthogonal $Q \in \mathbb{R}^{m\times m}$ and orthogonal $V \in \mathbb{R}^{n\times n}$ so that

$$Q^TA = \begin{bmatrix} R \\ 0 \end{bmatrix} \qquad Q^TBV = [\,0\,,\,S\,]$$

where $R \in \mathbb{R}^{n\times n}$ and $S \in \mathbb{R}^{m\times m}$ are upper triangular.

P12.1.11 Suppose $r \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, and $\delta > 0$. Show how to solve the problem

$$\min_{\substack{E \in \mathbb{R}^{m\times n} \\ \| E \|_F \leq \delta}} \| Ey - r \|_2$$

Repeat with "min" replaced by "max".

Notes and References for Sec. 12.1

Roughly speaking, regularization is a technique for transforming a poorly conditioned problem into a stable one. Quadratically constrained least squares is an important example. See

L. Eldén (1977). "Algorithms for the Regularization of Ill-Conditioned Least Squares Problems," *BIT 17*, 134–45.

References for cross-validation include

G.H. Golub, M. Heath, and G. Wahba (1979). "Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter," *Technometrics 21*, 215–23.

L. Eldén (1985). "A Note on the Computation of the Generalized Cross-Validation Function for Ill-Conditioned Least Squares Problems," *BIT 24*, 467–472.

The LSQI problem is discussed in

G.E. Forsythe and G.H. Golub (1965). "On the Stationary Values of a Second-Degree Polynomial on the Unit Sphere," *SIAM J. App. Math. 14,* 1050–68.
L. Eldén (1980). "Perturbation Theory for the Least Squares Problem with Linear Equality Constraints," *SIAM J. Num. Anal. 17,* 338–50.
W. Gander (1981). "Least Squares with a Quadratic Constraint," *Numer. Math. 36,* 291–307.
L. Eldén (1983). "A Weighted Pseudoinverse, Generalized Singular Values, and Constrained Least Squares Problems," *BIT 22* , 487–502.
G.W. Stewart (1984). "On the Asymptotic Behavior of Scaled Singular Value and QR Decompositions," *Math. Comp. 43,* 483–490.
G.H. Golub and U. von Matt (1991). "Quadratically Constrained Least Squares and Quadratic Problems," *Numer. Math. 59,* 561–580.
T.F. Chan, J.A. Olkin, and D. Cooley (1992). "Solving Quadratically Constrained Least Squares Using Black Box Solvers," *BIT 32,* 481–495.

Other computational aspects of the LSQI problem involve updating and the handling of banded and sparse problems. See

K. Schittkowski and J. Stoer (1979). "A Factorization Method for the Solution of Constrained Linear Least Squares Problems Allowing for Subsequent Data changes," *Numer. Math. 31,* 431–463.
D.P. O'Leary and J.A. Simmons (1981). "A Bidiagonalization-Regularization Procedure for Large Scale Discretizations of Ill-Posed Problems," *SIAM J. Sci. and Stat. Comp. 2,* 474–489.
Å. Björck (1984). "A General Updating Algorithm for Constrained Linear Least Squares Problems," *SIAM J. Sci. and Stat. Comp. 5,* 394–402.
L. Eldén (1984). "An Algorithm for the Regularization of Ill-Conditioned, Banded Least Squares Problems," *SIAM J. Sci. and Stat. Comp. 5,* 237–254.

Various aspects of the LSE problem are discussed and analyzed in

M.J.D. Powell and J.K. Reid (1968). "On Applying Householder's Method to Linear Least Squares Problems," *Proc. IFIP Congress,* pp. 122 26.
C. Van Loan (1985). "On the Method of Weighting for Equality Constrained Least Squares Problems," *SIAM J. Numer. Anal. 22,* 851–864.
J.L. Barlow, N.K. Nichols, and R.J. Plemmons (1988). "Iterative Methods for Equality Constrained Least Squares Problems," SIAM J. Sci. and Stat. Comp. 9, 892–906.
J.L. Barlow (1988). "Error Analysis and Implementation Aspects of Deferred Correction for Equality Constrained Least-Squares Problems," *SIAM J. Num. Anal. 25,* 1340–1358.
J.L. Barlow and S.L. Handy (1988). "The Direct Solution of Weighted and Equality Constrained Least-Squares Problems," *SIAM J. Sci. Stat. Comp. 9,* 704–716.
J.L. Barlow and U.B. Vemulapati (1992). "A Note on Deferred Correction for Equality Constrained Least Squares Problems," *SIAM J. Num. Anal. 29,* 249–256.
M. Wei (1992). "Perturbation Theory for the Rank-Deficient Equality Constrained Least Squares Problem," *SIAM J. Num. Anal. 29,* 1462–1481.
M. Wei (1992). "Algebraic Properties of the Rank-Deficient Equality-Constrained and Weighted Least Squares Problems," *Lin. Alg. and Its Applic. 161,* 27–44.
M. Gulliksson and P-Å. Wedin (1992). "Modifying the QR-Decomposition to Constrained and Weighted Linear Least Squares," *SIAM J. Matrix Anal. Appl. 13,* 1298–1313.
Å. Björck and C.C. Paige (1994). "Solution of Augmented Linear Systems Using Orthogonal Factorizations," *BIT 34,* 1–24.
M. Gulliksson (1994). "Iterative Refinement for Constrained and Weighted Linear Least Squares," *BIT 34,* 239–253.

M. Gulliksson (1995). "Backward Error Analysis for the Constrained and Weighted Linear Least Squares Problem When Using the Weighted QR Factorization," *SIAM J. Matrix. Anal. Appl. 13*, 675–687.

Generalized factorizations have an important bearing on generalized least squares problems.

C.C. Paige (1985). "The General Linear Model and the Generalized Singular Value Decomposition," *Lin. Alg. and Its Applic. 70*, 269–284.
C.C. Paige (1990). "Some Aspects of Generalized QR Factorization," in *Reliable Numerical Computations*, M. Cox and S. Hammarling (eds), Clarendon Press, Oxford.
E. Anderson, Z. Bai, and J. Dongarra (1992). "Generalized QR Factorization and Its Applications," *Lin. Alg. and Its Applic. 162/163/164*, 243 271.

12.2 Subset Selection Using the SVD

As described in §5.5, the rank-deficient LS problem min $\| Ax - b \|_2$ can be approached by approximating the minimum norm solution

$$x_{LS} = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i \qquad r = \text{rank}(A)$$

with

$$x_{\tilde{r}} = \sum_{i=1}^{\tilde{r}} \frac{u_i^T b}{\sigma_i} v_i \qquad \tilde{r} \le r$$

where

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T \tag{12.2.1}$$

is the SVD of A and $\tilde{r}$ is some numerically determined estimate of r. Note that $x_{\tilde{r}}$ minimizes $\| A_{\tilde{r}} x - b \|_2$ where

$$A_{\tilde{r}} = \sum_{i=1}^{\tilde{r}} \sigma_i u_i v_i^T$$

is the closest matrix to A that has rank $\tilde{r}$. See Theorem 2.5.3.

Replacing A by $A_{\tilde{r}}$ in the LS problem amounts to filtering the small singular values and can make a great deal of sense in those situations where A is derived from noisy data. In other applications, however, rank deficiency implies redundancy among the factors that comprise the underlying model. In this case, the model-builder may not be interested in a predictor such as $A_{\tilde{r}} x_{\tilde{r}}$ that involves all n redundant factors. Instead, a predictor Ay may be sought where y has at most $\tilde{r}$ nonzero components. The position of the nonzero entries determines which columns of A, i.e., which factors in the model, are to be used in approximating the observation vector b. How to pick these columns is the problem of *subset selection* and is the subject of this section.

The contents of this section depends heavily upon §2.6 and Chapter 5.

12.2.1 QR with Column Pivoting

QR with column pivoting can be regarded as a method for selecting an independent subset of A's columns from which b might be predicted. Suppose we apply Algorithm 5.4.1 to $A \in \mathbb{R}^{m\times n}$ and compute an orthogonal Q and a permutation Π such that $R = Q^T A\Pi$ is upper triangular. If $R(1{:}\tilde{r}, 1{:}\tilde{r})z = \tilde{b}(1{:}\tilde{r})$ where $\tilde{b} = Q^T b$ and we set

$$y = \Pi \begin{bmatrix} z \\ 0 \end{bmatrix},$$

then Ay is an approximate LS predictor of b that involves the first $\tilde{r}$ columns of $A\Pi$.

12.2.2 Using the SVD

Although QR with column pivoting is a fairly reliable way to handle near rank deficiency, the SVD is sometimes preferable for reasons discussed in §5.5. We therefore describe an SVD-based subset selection procedure due to Golub, Klema, and Stewart (1976) that proceeds as follows:

- Compute the SVD $A = U\Sigma V^T$ and use it to determine a rank estimate $\tilde{r}$.
- Calculate a permutation matrix P such that the columns of the matrix $B_1 \in \mathbb{R}^{m\times\tilde{r}}$ in $AP = [\, B_1 \; B_2 \,]$ are "sufficiently independent."
- Predict b with the vector Ay where $y = P \begin{bmatrix} z \\ 0 \end{bmatrix}$ and $z \in \mathbb{R}^{\tilde{r}}$ minimizes $\| B_1 z - b \|_2$

The second step is key. Since

$$\min_{z \in \mathbb{R}^{\tilde{r}}} \| B_1 z - b \|_2 \;=\; \| Ay - b \|_2 \;\geq\; \min_{x \in \mathbb{R}^{n}} \| Ax - b \|_2$$

it can be argued that the permutation P should be chosen to make the residual $(I - B_1 B_1^{+})b$ as small as possible. Unfortunately, such a solution procedure can be unstable. For example, if

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1+\epsilon & 1 \\ 0 & 0 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix},$$

$\tilde{r} = 2$, and $P = I$, then $\min \| B_1 z - b \|_2 = 0$, but $\| B_1^{+} b \|_2 = O(1/\epsilon)$. On the other hand, any proper subset involving the third column of A is strongly independent but renders a much worse residual.

This example shows that there can be a trade-off between the independence of the chosen columns and the norm of the residual that they render. How to proceed in the face of this trade-off requires additional mathematical machinery in the form of useful bounds on $\sigma_{\tilde{r}}(B_1)$, the smallest singular value of B_1.

Theorem 12.2.1 *Let the SVD of $A \in \mathbb{R}^{m\times n}$ be given by (12.2.1), and define the matrix $B_1 \in \mathbb{R}^{m\times\tilde{r}}$, $\tilde{r} \le \text{rank}(A)$, by*

$$AP = \underset{\tilde{r}\quad\;\; n-\tilde{r}}{[\; B_1 \quad B_2 \;]}$$

where $P \in \mathbb{R}^{n\times n}$ is a permutation. If

$$P^TV = \begin{bmatrix} \tilde{V}_{11} & \tilde{V}_{12} \\ \tilde{V}_{21} & \tilde{V}_{22} \end{bmatrix} \begin{matrix} \tilde{r} \\ n-\tilde{r} \end{matrix} \qquad (12.2.2)$$
$$\qquad\quad \tilde{r} \quad n-\tilde{r}$$

and $\tilde{V}_{11}$ is nonsingular, then

$$\frac{\sigma_{\tilde{r}}(A)}{\| \tilde{V}_{11}^{-1} \|_2} \le \sigma_{\tilde{r}}(B_1) \le \sigma_{\tilde{r}}(A).$$

Proof. The upper bound follows from the minimax characterization of singular values given in §8.6.1

To establish the lower bound, partition the diagonal matrix of singular values as follows:

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{matrix} \tilde{r} \\ m-\tilde{r} \end{matrix} .$$
$$\qquad \tilde{r} \quad n-\tilde{r}$$

If $w \in \mathbb{R}^{\tilde{r}}$ is a unit vector with the property that $\| B_1 w \|_2 = \sigma_{\tilde{r}}(B_1)$, then

$$\begin{aligned} \sigma_{\tilde{r}}(B_1)^2 &= \| B_1 w \|_2^2 = \left\| U\Sigma V^T P \begin{bmatrix} w \\ 0 \end{bmatrix} \right\|_2^2 \\ &= \| \Sigma_1 \tilde{V}_{11}^T w \|_2^2 + \| \Sigma_2 \tilde{V}_{12}^T w \|_2^2 . \end{aligned}$$

The theorem now follows because $\| \Sigma_1 \tilde{V}_{11}^T w \|_2 \ge \sigma_{\tilde{r}}(A)/\| \tilde{V}_{11}^{-1} \|_2$. □

This result suggests that in the interest of obtaining a sufficiently independent subset of columns, we choose the permutation P such that the resulting $\tilde{V}_{11}$ submatrix is as well-conditioned as possible. A heuristic solution to this problem can be obtained by computing the QR with column-pivoting factorization of the matrix $[\; V_{11}^T \; V_{21}^T \;]$, where

$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \begin{matrix} \tilde{r} \\ n-\tilde{r} \end{matrix}$$
$$\qquad \tilde{r} \quad n-\tilde{r}$$

is a partitioning of the matrix V in (12.2.1). In particular, if we apply QR with column pivoting (Algorithm 5.4.1) to compute

$$Q^T[\,V_{11}^T\;V_{21}^T\,]P \;=\; [\;\underset{\tilde{r}}{R_{11}}\quad \underset{n-\tilde{r}}{R_{12}}\;]$$

where Q is orthogonal, P is a permutation matrix, and R_{11} is upper triangular, then (12.2.2) implies:

$$\begin{bmatrix} \tilde{V}_{11} \\ \tilde{V}_{21} \end{bmatrix} = P^T \begin{bmatrix} V_{11} \\ V_{21} \end{bmatrix} = \begin{bmatrix} R_{11}^T Q^T \\ R_{12}^T Q^T \end{bmatrix}.$$

Note that R_{11} is nonsingular and that $\| \tilde{V}_{11}^{-1} \|_2 = \| R_{11}^{-1} \|_2$. Heuristically, column pivoting tends to produce a well-conditioned R_{11}, and so the overall process tends to produce a well-conditoned $\tilde{V}_{11}$. Thus we obtain

Algorithm 12.2.1 Given $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ the following algorithm computes a permutation P, a rank estimate $\tilde{r}$, and a vector $z \in \mathbb{R}^{\tilde{r}}$ such that the first $\tilde{r}$ columns of $B = AP$ are independent and such that $\| B(:,1{:}\tilde{r})z - b \|_2$ is minimized.

> Compute the SVD $U^T AV = \text{diag}(\sigma_1, \ldots, \sigma_n)$ and save V.
> Determine $\tilde{r} \le \text{rank}(A)$.
> Apply QR with column pivoting: $Q^T V(:,1{:}\tilde{r})^T P = [\,R_{11}\; R_{12}\,]$ and set
> $\quad AP = [\,B_1\; B_2\,]$ with $B_1 \in \mathbb{R}^{m\times\tilde{r}}$ and $B_2 \in \mathbb{R}^{m\times(n-\tilde{r})}$.
> Determine $z \in \mathbb{R}^{\tilde{r}}$ such that $\| b - B_1 z \|_2 = \min$.

Example 12.2.1 Let

$$A = \begin{bmatrix} 3 & 4 & 1.0001 \\ 7 & 4 & -3.0002 \\ 2 & 5 & 2.9999 \\ -1 & 4 & 5.0003 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

A is close to being rank 2 in the sense that $\sigma_3(A) \approx .0001$. Setting $\tilde{r} = 2$ in Algorithm 12.2.1 leads to $x = [0\;\; 0.2360\;\; -0.0085]^T$ with $\| Ax - b \|_2 = .1966$. (The permutation P is given by $P = [e_3\;\; e_2\;\; e_1]$.) Note that $x_{LS} = [828.1056\;\; -827.8569\;\; 828.0536]^T$ with minimum residual $\| Ax_{LS} - b \|_2 = 0.0343$.

12.2.3 More on Column Independence vs. Residual

We return to the discussion of the trade-off between column independence and norm of the residual. In particular, to assess the above method of subset selection, we need to examine the residual of the vector y that it produces $r_y = b - Ay = b - B_1 z = (I - B_1 B_1^+)b$. Here, $B_1 = B(:,1{:}\tilde{r})$ with

$B = AP$. To this end, it is appropriate to compare r_y with $r_{x_{\tilde{r}}} = b - Ax_{\tilde{r}}$ since we are regarding A as a rank-$\tilde{r}$ matrix and since $x_{\tilde{r}}$ solves the nearest rank-$\tilde{r}$ LS problem, namely, min $\| A_{\tilde{r}}x - b \|_2$.

Theorem 12.2.2 *If r_y and $r_{x_{\tilde{r}}}$ are defined as above and if $\tilde{V}_{11}$ is the leading r-by-r principal submatrix of P^TV, then*

$$\| r_{x_{\tilde{r}}} - r_y \|_2 \leq \frac{\sigma_{\tilde{r}+1}(A)}{\sigma_{\tilde{r}}(A)} \| \tilde{V}_{11}^{-1} \|_2 \| b \|_2$$

Proof. Note that $r_{x_{\tilde{r}}} = (I - U_1U_1^T)b$ and $r_y = (I - Q_1Q_1^T)b$ where

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \quad \begin{matrix} \tilde{r} & m - \tilde{r} \end{matrix}$$

is a partitioning of the matrix U in (12.2.1) and where $Q_1 = B_1(B_1^TB_1)^{-1/2}$. Using Theorem 2.6.1 we obtain

$$\| r_{x_{\tilde{r}}} - r_y \|_2 \leq \| U_1U_1^T - Q_1Q_1^T \|_2 \| b \|_2 = \| U_2^TQ_1 \|_2 \| b \|_2$$

while Theorem 12.2.1 permits us to conclude that

$$\begin{aligned} \| U_2^TQ_1 \|_2 &\leq \| U_2^TB_1 \|_2 \| (B_1^TB_1)^{-1/2} \|_2 \leq \sigma_{\tilde{r}+1}(A)\frac{1}{\sigma_{\tilde{r}}(B_1)} \\ &\leq \frac{\sigma_{\tilde{r}+1}(A)}{\sigma_{\tilde{r}}(A)} \| \tilde{V}_{11}^{-1} \|_2. \ \square \end{aligned}$$

Noting that

$$\| r_{x_{\tilde{r}}} - r_y \|_2 = \left\| B_1y - \sum_{i=1}^{r}(u_i^Tb)u_i \right\|_2$$

we see that Theorem 12.2.2 sheds light on how well B_1y can predict the "stable" component of b, i.e., U_1^Tb. Any attempt to approximate U_2^Tb can lead to a large norm solution. Moreover, the theorem says that if $\sigma_{\tilde{r}+1}(A) \ll \sigma_{\tilde{r}}(A)$, then any reasonably independent subset of columns produces essentially the same-sized residual. On the other hand, if there is no well-defined gap in the singular values, then the determination of $\tilde{r}$ becomes difficult and the entire subset selection problem more complicated.

Problems

P12.2.1 Suppose $A \in \mathbb{R}^{m\times n}$ and that $\| u^TA \|_2 = \sigma$ with $u^Tu = 1$. Show that if $u^T(Ax - b) = 0$ for $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$, then $\| x \|_2 \geq |u^Tb|/\sigma$.

P12.2.2 Show that if $B_1 \in \mathbb{R}^{m\times k}$ is comprised of k columns from $A \in \mathbb{R}^{m\times n}$ then $\sigma_k(B_1) \leq \sigma_k(A)$.

P12.2.3 In equation (12.2.2) we know that the matrix

$$P^TV = \begin{bmatrix} \tilde{V}_{11} & \tilde{V}_{12} \\ \tilde{V}_{21} & \tilde{V}_{22} \end{bmatrix} \begin{matrix} \tilde{r} \\ n - \tilde{r} \end{matrix}$$
$$\qquad\quad \begin{matrix} \tilde{r} & n - \tilde{r} \end{matrix}$$

is orthogonal. Thus, $\| \tilde{V}_{11}^{-1} \|_2 = \| \tilde{V}_{22}^{-1} \|_2$ from the CS decomposition (Theorem 2.6.3). Show how to compute P by applying the QR with column pivoting algorithm to $[\tilde{V}_{22}^T \; \tilde{V}_{12}^T]$. (For $\tilde{r} > n/2$, this procedure would be more economical than the technique discussed in the text.) Incorporate this observation in Algorithm 12.2.1.

Notes and References for Sec. 12.2

The material in this section is derived from

G.H. Golub, V. Klema and G.W. Stewart (1976). "Rank Degeneracy and Least Squares Problems," Technical Report TR-456, Department of Computer Science, University of Maryland, College Park, MD.

A subset selection procedure based upon the total least squares fitting technique of §12.3 is given in

S. Van Huffel and J. Vandewalle (1987). "Subset Selection Using the Total Least Squares Approach in Collinearity Problems with Errors in the Variables," *Lin. Alg. and Its Applic.* *88/89*, 695–714.

The literature on subset selection is vast and we refer the reader to

H. Hotelling (1957). "The Relations of the Newer Multivariate Statistical Methods to Factor Analysis," *Brit. J. Stat. Psych.* *10*, 69–79.

12.3 Total Least Squares

The problem of minimizing $\| D(Ax - b) \|_2$ where $A \in \mathbb{R}^{m \times n}$, and $D = \text{diag}(d_1, \ldots, d_m)$ is nonsingular can be recast as follows:

$$\min_{b+r \,\in\, \text{range}(A)} \| Dr \|_2 \qquad r \in \mathbb{R}^m . \tag{12.3.1}$$

In this problem, there is a tacit assumption that the errors are confined to the "observation" b. When error is also present in the "data" A, then it may be more natural to consider the problem

$$\min_{b+r \,\in\, \text{range}(A+E)} \| D[\, E \,,\, r \,]T \|_F \qquad E \in \mathbb{R}^{m \times n},\; r \in \mathbb{R}^m \tag{12.3.2}$$

where $D = \text{diag}(d_1, \ldots, d_m)$ and $T = \text{diag}(t_1, \ldots, t_{n+1})$ are nonsingular. This problem, discussed in Golub and Van Loan (1980), is referred to as the *total least squares* (TLS) problem.

If a minimizing $[\, E_0 \,,\, r_0 \,]$ can be found for (12.3.2), then any x satisfying $(A + E_0)x = b + r_0$ is called a TLS solution. However, it should be realized that (12.3.2) may fail to have a solution altogether. For example, if

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \; b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \; D = I_3,\; T = I_3, \text{ and } E_\epsilon = \begin{bmatrix} 0 & 0 \\ 0 & \epsilon \\ 0 & \epsilon \end{bmatrix}$$

then for all $\epsilon > 0$, $b \in \text{ran}(A + E_\epsilon)$. However, there is no smallest value of $\| [E, r] \|_F$ for which $b + r \in \text{ran}(A + E)$.

A generalization of (12.3.2) results if we allow multiple right-hand sides. In particular, if $B \in \mathbb{R}^{m \times k}$, then we have the problem

$$\min_{\text{range}(B+R) \subseteq \text{range}(A+E)} \| D[E, R]T \|_F \qquad (12.3.3)$$

where $E \in \mathbb{R}^{m \times n}$ and $R \in \mathbb{R}^{m \times k}$ and the matrices $D = \text{diag}(d_1, \ldots, d_m)$ and $T = \text{diag}(t_1, \ldots, t_{n+k})$ are nonsingular. If $[E_0, R_0]$ solves (12.3.3), then any $X \in \mathbb{R}^{n \times k}$ that satisfies $(A + E_0)X = (B + R_0)$ is said to be a TLS solution to (12.3.3).

In this section we discuss some of the mathematical properties of the total least squares problem and show how it can be solved using the SVD. Chapter 5 is the only prerequisite. A very detailed treatment of the TLS problem is given in the monograph by Van Huffel and Vanderwalle (1991).

12.3.1 Mathematical Background

The following theorem gives conditions for the uniqueness and existence of a TLS solution to the multiple right-hand side problem.

Theorem 12.3.1 *Let A, B, D, and T be as above and assume $m \geq n+k$. Let*

$$C = D[A, B]T = [\underset{n}{C_1} \quad \underset{k}{C_2}]$$

have SVD $U^TCV = \text{diag}(\sigma_1, \ldots, \sigma_{n+k}) = \Sigma$ where U, V, and Σ are partitioned as follows:

$$U = [\underset{n}{U_1} \quad \underset{k}{U_2}] \qquad V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \begin{matrix} n \\ k \end{matrix} \quad (\text{columns } n, k)$$

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{matrix} n \\ k \end{matrix} \quad (\text{columns } n, k)$$

If $\sigma_n(C_1) > \sigma_{n+1}(C)$, then the matrix $[E_0, R_0]$ defined by

$$D[E_0, R_0]T = -U_2\Sigma_2[V_{12}^T, V_{22}^T] \qquad (12.3.4)$$

solves (12.3.3). If $T_1 = \text{diag}(t_1, \ldots, t_n)$ and $T_2 = \text{diag}(t_{n+1}, \ldots, t_{n+k})$ then the matrix

$$X_{TLS} = -T_1V_{12}V_{22}^{-1}T_2^{-1}$$

exists and is the unique solution to $(A + E_0)X = B + R_0$.

Proof. We first establish two results that follow from the assumption $\sigma_n(C_1) > \sigma_{n+1}(C)$. From the equation $CV = U\Sigma$ we have $C_1V_{12}+C_2V_{22} = U_2\Sigma_2$. We wish to show that V_{22} is nonsingular. Suppose $V_{22}x = 0$ for some unit 2-norm x. It follows from $V_{12}^TV_{12} + V_{22}^TV_{22} = I$ that $\| V_{12}x \|_2 = 1$. But then

$$\sigma_{n+1}(C) \geq \| U_2\Sigma_2 x \|_2 = \| C_1V_{12}x \|_2 \geq \sigma_n(C_1),$$

a contradiction. Thus, the submatix V_{22} is nonsingular.

The other fact that follows from $\sigma_n(C_1) > \sigma_{n+1}(C)$ concerns the strict separation of $\sigma_n(C)$ and $\sigma_{n+1}(C)$. From Corollary 8.3.3, we have $\sigma_n(C) \geq \sigma_n(C_1)$ and so $\sigma_n(C) \geq \sigma_n(C_1) > \sigma_{n+1}(C)$.

Now we are set to prove the theorem. If $\text{ran}(B + R) \subset \text{ran}(A + E)$, then there is an X (n-by-k) so $(A + E)X = B + R$, i.e.,

$$\{ D[A, B]T + D[E, R]T \} T^{-1} \begin{bmatrix} X \\ -I_k \end{bmatrix} = 0. \qquad (12.3.5)$$

Thus, the matrix in curly brackets has, at most, rank n. By following the argument in Theorem 2.5.3, it can be shown that

$$\| D[E, R]T \|_F \geq \sum_{i=n+1}^{n+k} \sigma_i(C)^2$$

and that the lower bound is realized by setting $[E, R] = [E_0, R_0]$. The inequality $\sigma_n(C) > \sigma_{n+1}(C)$ ensures that $[E_0, R_0]$ is the unique minimizer. The null space of

$$\{D[A, B]T + D[E_0, R_0]T\} = U_1\Sigma_1[V_{11}^T\ V_{21}^T]$$

is the range of $\begin{bmatrix} V_{12} \\ V_{22} \end{bmatrix}$. Thus, from (12.3.5)

$$T^{-1}\begin{bmatrix} X \\ -I_k \end{bmatrix} = \begin{bmatrix} V_{12} \\ V_{22} \end{bmatrix} S$$

for some k-by-k matrix S. From the equations $T_1^{-1}X = V_{12}S$ and $-T_2^{-1} = V_{22}S$ we see that $S = -V_{22}^{-1}T_2^{-1}$. Thus, we must have

$$X = T_1V_{12}S = -T_1V_{12}V_{22}^{-1}T_2^{-1} = X_{TLS}. \ \square$$

If $\sigma_n(C) = \sigma_{n+1}(C)$, then the TLS problem may still have a solution, although it may not be unique. In this case, it may be desirable to single out a "minimal norm" solution. To this end, consider the τ-norm defined on $\mathbb{R}^{n\times k}$ by $\| Z \|_\tau = \| T_1^{-1}ZT_2 \|_2$. If X is given by (12.3.5), then from the CS decomposition (Theorem 2.6.3) we have

$$\| X \|_\tau^2 = \| V_{12}V_{22}^{-1} \|_2^2 = \frac{1 - \sigma_k(V_{22})^2}{\sigma_k(V_{22})^2}.$$

This suggests choosing V in Theorem 12.3.1 so that $\sigma_k(V_{22})$ is maximized.

12.3.2 Computations for the k=1 Case

We show how to maximize V_{22} in the important $k = 1$ case. Suppose the singular values of C satisfy $\sigma_{n-p} > \sigma_{n-p+1} = \cdots = \sigma_{n+1}$ and let $V = [\, v_1, \ldots, v_{n+1} \,]$ be a column partitioning of V. If $\tilde{Q}$ is a Householder matrix such that

$$V(:, n+1-p{:}n+1)\tilde{Q} \;=\; \begin{bmatrix} W & z \\ 0 & \alpha \end{bmatrix} \begin{matrix} n \\ 1 \end{matrix}$$
$$\qquad\qquad\qquad\qquad\quad p \quad 1$$

then $\begin{bmatrix} z \\ \alpha \end{bmatrix}$ has the largest $(n+1)$-st component of all the vectors in span$\{v_{n+1-p}, \ldots, v_{n+1}\}$. If $\alpha = 0$, the TLS problem has no solution. Otherwise $x_{TLS} = -T_1 z/(t_{n+1}\alpha)$. Moreover,

$$\begin{bmatrix} I_{n-1} & 0 \\ 0 & Q \end{bmatrix} U^T(D[\, A\,, b\,]T)V \begin{bmatrix} I_{n-p} & 0 \\ 0 & \tilde{Q} \end{bmatrix} \;=\; \Sigma$$

and so

$$D[\, E_0\,, r_0\,]T \;=\; -D[\, A\,, b\,]T \begin{bmatrix} z \\ \alpha \end{bmatrix} [\, z^T \; \alpha\,].$$

Overall, we have the following algorithm:

Algorithm 12.3.1 Given $A \in \mathbb{R}^{m\times n}$ $(m > n)$, $b \in \mathbb{R}^m$, and nonsingular $D = \text{diag}(d_1, \ldots, d_m)$ and $T = \text{diag}(t_1, \ldots, t_{n+1})$, the following algorithm computes (if possible) a vector $x_{TLS} \in \mathbb{R}^n$ such that $(A + E_0)x = (b + r_0)$ and $\| \, D[\, E_0\,, r_0\,]T \,\|_F$ is minimal.

Compute the SVD $U^T(D[\, A\,, b\,]T)V \;=\; \text{diag}(\sigma_1, \ldots, \sigma_{n+1})$. Save V.
Determine p such that $\sigma_1 \geq \cdots \geq \sigma_{n-p} > \sigma_{n-p+1} = \cdots = \sigma_{n+1}$.
Compute a Householder matrix P such that if $\tilde{V} = VP$, then
    $\tilde{V}(n+1, n-p+1{:}n) = 0$

if $\tilde{v}_{n+1,n+1} \neq 0$
    for $i = 1{:}n$
        $x_i = -t_i\tilde{v}_{i,n+1}/(t_{n+1}\tilde{v}_{n+1,n+1})$
    end
end

This algorithm requires about $2mn^2 + 12n^3$ flops and most of these are associated with the SVD computation.

Example 12.3.1 The TLS problem $\displaystyle\min_{(a+e)x=b+r} \| \, [\, e\,, r\,] \,\|_F$ where $a = [1, 2, 3, 4]^T$ and $b = [2.01, 3.99, 5.80, 8.30]^T$ has solution $x_{TLS} = 2.0212$, $e = [-.0045, -.0209, -.1048, .0855]^T$, and $r = [.0022, .0103, .0519, -.0423]^T$. Note that for this data $x_{LS} = 2.0197$.

12.3.3 Geometric Interpretation

It can be shown that the TLS solution x_{TLS} minimizes

$$\psi(x) = \sum_{i=1}^{m} d_i^2 \frac{|a_i^T x - b_i|^2}{x^T T_1^{-2} x + t_{n+1}^{-2}}$$

where a_i^T is ith row of A and b_i is the ith component of b. A geometrical interpretation of the TLS problem is made possible by this observation. Indeed,

$$\frac{|a_i^T x - b_i|^2}{x^T T_1^{-2} x + t_{n+1}^{-2}}$$

is the square of the distance from $\begin{bmatrix} a_i \\ b_i \end{bmatrix} \in \mathbb{R}^{n+1}$ to the nearest point in the subspace

$$P_x = \left\{ \begin{bmatrix} a \\ b \end{bmatrix} : a \in \mathbb{R}^n,\ b \in \mathbb{R},\ b = x^T a \right\}$$

where distance in $\mathbb{R}^{n+1}$ is measured by the norm $\| z \| = \| Tz \|_2$. A great deal has been written about this kind of fitting. See Pearson (1901) and Madansky (1959).

Problems

P12.3.1 Consider the TLS problem (12.3.2) with nonsingular D and T. (a) Show that if $\text{rank}(A) < n$, then (12.3.2) has a solution if and only if $b \in \text{ran}(A)$. (b) Show that if $\text{rank}(A) = n$, then (12.3.2) has no solution if $A^T D^2 b = 0$ and $|t_{n+1}| \| Db \|_2 \geq \sigma_n(DAT_1)$ where $T_1 = \text{diag}(t_1, \ldots, t_n)$.

P12.3.2 Show that if $C = D[\, A\, ,\, b\,]T = [\, A_1\, ,\, d\,]$ and $\sigma_n(C) > \sigma_{n+1}(C)$, then the TLS solution x satisfies $(A_1^T A_1 - \sigma_{n+1}(C)^2 I)x = A_1^T d$.

P12.3.3 Show how to solve (12.3.2) with the added constraint that the first p columns of the minimizing E are zero.

Notes and References for Sec. 12.3

This section is based upon

G.H. Golub and C.F. Van Loan (1980). "An Analysis of the Total Least Squares Problem," *SIAM J. Num. Anal.* *17*, 883–93.

The bearing of the SVD on the TLS problem is set forth in

G.H. Golub and C. Reinsch (1970). "Singular Value Decomposition and Least Squares Solutions," *Numer. Math.* *14*, 403–420.

G.H. Golub (1973). "Some Modified Matrix Eigenvalue Problems," *SIAM Review 15*, 318–334.

The most detailed study of the TLS problem is

S. Van Huffel and J. Vandewalle (1991). *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM Publications, Philadelphia.

If some of the columns of A are known exactly then it is sensible to force the TLS perturbation matrix E to be zero in the same columns. Aspects of this constrained TLS problem are discussed in

J.W. Demmel (1987). "The Smallest Perturbation of a Submatrix which Lowers the Rank and Constrained Total Least Squares Problems, *SIAM J. Numer. Anal.* 24, 199–206.
S. Van Huffel and J. Vandewalle (1988). "The Partial Total Least Squares Algorithm," *J. Comp. and App. Math. 21*, 333–342.
S. Van Huffel and J. Vandewalle (1988). "Analysis and Solution of the Nongeneric Total Least Squares Problem," *SIAM J. Matrix Anal. Appl. 9*, 360–372.
S. Van Huffel and J. Vandewalle (1989). "Analysis and Properties of the Generalized Total Least Squares Problem $AX \approx B$ When Some or All Columns in A are Subject to Error," *SIAM J. Matrix Anal. Appl. 10*, 294–315.
S. Van Huffel and H. Zha (1991). "The Restricted Total Least Squares Problem: Formulation, Algorithm, and Properties," *SIAM J. Matrix Anal. Appl. 12*, 292–309.
S. Van Huffel (1992). "On the Significance of Nongeneric Total Least Squares Problems," *SIAM J. Matrix Anal. Appl. 13*, 20–35.
M. Wei (1992). "The Analysis for the Total Least Squares Problem with More than One Solution," *SIAM J. Matrix Anal. Appl. 13*, 746–763.
S. Van Huffel and H. Zha (1993). "An Efficient Total Least Squares Algorithm Based On a Rank-Revealing Two-Sided Orthogonal Decomposition," *Numerical Algorithms 4*, 101 133.
C.C. Paige and M. Wei (1993). "Analysis of the Generalized Total Least Squares Problem $AX = B$ when Some of the Columns are Free of Error," *Numer. Math. 65*, 177–202.
R.D. Fierro and J.R. Bunch (1994). "Collinearity and Total Least Squares," *SIAM J. Matrix Anal. Appl. 15*, 1167 1181.

Other references concerned with least squares fitting when there are errors in the data matrix include

K. Pearson (1901). "On Lines and Planes of Closest Fit to Points in Space," *Phil. Mag. 2*, 559–72.
A. Wald (1940). "The Fitting of Straight Lines if Both Variables are Subject to Error," *Annals of Mathematical Statistics 11*, 284–300.
A. Madansky (1959). "The Fitting of Straight Lines When Both Variables Are Subject to Error," *J. Amer. Stat. Assoc. 54*, 173–205.
I. Linnik (1961). *Method of Least Squares and Principles of the Theory of Observations*, Pergamon Press, New York.
W.G. Cochrane (1968). "Errors of Measurement in Statistics," *Technometrics 10*, 637–66.
R.F. Gunst, J.T. Webster, and R.L. Mason (1976). "A Comparison of Least Squares and Latent Root Regression Estimators," *Technometrics 18*, 75–83.
G.W. Stewart (1977c). "Sensitivity Coefficients for the Effects of Errors in the Independent Variables in a Linear Regression," Technical Report TR-571, Department of Computer Science, University of Maryland, College Park, MD.
A. Van der Sluis and G.W. Veltkamp (1979). "Restoring Rank and Consistency by Orthogonal Projection," *Lin. Alg. and Its Applic. 28*, 257–78.

12.4 Computing Subspaces with the SVD

It is sometimes necessary to investigate the relationship between two given subspaces. How close are they? Do they intersect? Can one be "rotated" into the other? And so on. In this section we show how questions like these can be answered using the singular value decomposition. Knowledge of Chapter 5 and §8.6 are assumed.

12.4.1 Rotation of Subspaces

Suppose $A \in \mathbb{R}^{m\times p}$ is a data matrix obtained by performing a certain set of experiments. If the same set of experiments is performed again, then a different data matrix, $B \in \mathbb{R}^{m\times p}$, is obtained. In the *orthogonal Procrustes problem* the possibility that B can be rotated into A is explored by solving the following problem:

$$\text{minimize } \| A - BQ \|_F \qquad \text{subject to } Q^TQ = I_p. \tag{12.4.1}$$

Recall that the trace of a matrix is the sum of its diagonal entries and thus, $\text{tr}(C^TC) = \| C \|_F^2$. It follows that if $Q \in \mathbb{R}^{p\times p}$ is orthogonal, then

$$\| A - BQ \|_F^2 = \text{tr}(A^TA) + \text{tr}(B^TB) - 2\,\text{tr}(Q^TB^TA).$$

Thus, (12.4.1) is equivalent to the problem of maximizing $\text{tr}(Q^TB^TA)$.

The maximizing Q can be found by calculating the SVD of B^TA. Indeed, if $U^T(B^TA)V = \Sigma = \text{diag}(\sigma_1,\ldots,\sigma_p)$ is the SVD of this matrix and we define the orthogonal matrix Z by $Z = V^TQ^TU$, then

$$\text{tr}(Q^TB^TA) = \text{tr}(Q^TU\Sigma V^T) = \text{tr}(Z\Sigma) = \sum_{i=1}^{p} z_{ii}\sigma_i \le \sum_{i=1}^{p}\sigma_i.$$

Clearly, the upper bound is attained by setting $Q = UV^T$ for then $Z = I_p$. This gives the following algorithm:

Algorithm 12.4.1 Given A and B in $\mathbb{R}^{m\times p}$, the following algorithm finds an orthogonal $Q \in \mathbb{R}^{p\times p}$ such that $\| A - BQ \|_F$ is minimum.

$C = B^TA$
Compute the SVD $U^TCV = \Sigma$. Save U and V.
$Q = UV^T$.

The solution matrix Q is the orthogonal polar factor of B^TA. See §4.2.10.

Example 12.4.1

$$Q = \begin{bmatrix} .9999 & -.0126 \\ .0126 & .9999 \end{bmatrix} \quad \text{minimizes} \quad \left\| \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix} Q - \begin{bmatrix} 1.2 & 2.1 \\ 2.9 & 4.3 \\ 5.2 & 6.1 \\ 6.8 & 8.1 \end{bmatrix} \right\|_F .$$

12.4.2 Intersection of Null Spaces

Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$ be given, and consider the problem of finding an orthonormal basis for null$(A) \cap$ null(B). One approach is to compute the null space of the matrix

$$C = \begin{bmatrix} A \\ B \end{bmatrix}$$

since $Cx = 0 \Leftrightarrow x \in \text{null}(A) \cap \text{null}(B)$. However, a more economical procedure results if we exploit the following theorem.

Theorem 12.4.1 *Suppose* $A \in \mathbb{R}^{m\times n}$ *and let* $\{z_1, \ldots, z_t\}$ *be an orthonormal basis for* null(A). *Define* $Z = [\, z_1, \ldots, z_t \,]$ *and let* $\{w_1, \ldots, w_q\}$ *be an orthonormal basis for* null(BZ) *where* $B \in \mathbb{R}^{p\times n}$. *If* $W = [\, w_1, \ldots, w_q \,]$, *then the columns of* ZW *form an orthonormal basis for* null$(A) \cap$ null(B).

Proof. Since $AZ = 0$ and $(BZ)W = 0$, we clearly have ran$(ZW) \subset$ null$(A) \cap$ null(B). Now suppose x is in both null(A) and null(B). It follows that $x = Za$ for some $0 \neq a \in \mathbb{R}^t$. But since $0 = Bx = BZa$, we must have $a = Wb$ for some $b \in \mathbb{R}^q$. Thus, $x = ZWb \in$ ran(ZW). □

When the SVD is used to compute the orthonormal bases in this theorem we obtain the following procedure:

Algorithm 12.4.2 Given $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$, the following algorithm computes and integer s and a matrix $Y = [\, y_1, \ldots, y_s \,]$ having orthonormal columns which span null$(A) \cap$ null(B). If the intersection is trivial then $s = 0$.

```
Compute the SVD U_A^T A V_A = diag(σ_i). Save V_A and set
        r = rank(A).
if r < n
    C = B V_A(:, r+1:n)
    Compute the SVD U_C^T C V_C = diag(γ_i). Save V_C and set
        q = rank(C).
    if q < n − r
        s = n − r − q
        Y = V_A(:, r+1:n) V_C(:, q+1:n − r)
    else
        s = 0
    end
else
    s = 0
end
```

The amount of work required by this algorithm depends upon the relative sizes of m, n, p, and r.

We mention that a practical implementation of this algorithm requires a means for deciding when a computed singular value $\hat{\sigma}_i$ is negligible. The use of a tolerance δ for this purpose (e.g. $\hat{\sigma}_i < \delta \Rightarrow \hat{\sigma}_i = 0$) implies that the columns of the computed $\hat{Y}$ "almost" define a common null space of A and B in the sense that $\| A\hat{Y} \|_2 \approx \| B\hat{Y} \|_2 \approx \delta$.

Example 12.4.2 If

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 4 & 2 & 0 \\ 2 & 1 & 0 \\ 6 & 3 & 0 \end{bmatrix}$$

then $\text{null}(A) \cap \text{null}(B) = \text{span}\{x\}$, where $x = [1\ -2\ -3]^T$. Applying Algorithm 12.4.2 we find

$$V_{2A}V_{2C} = \begin{bmatrix} -.8165 & .0000 \\ -.4082 & .7071 \\ .4082 & .7071 \end{bmatrix} \begin{bmatrix} -.3273 \\ -.9449 \end{bmatrix} \approx \begin{bmatrix} .2673 \\ -.5345 \\ -.8018 \end{bmatrix} \approx .2673 \begin{bmatrix} 1 \\ -2 \\ -3 \end{bmatrix}.$$

12.4.3 Angles Between Subspaces

Let F and G be subspaces in $\mathbb{R}^m$ whose dimensions satisfy

$$p = \dim(F) \geq \dim(G) = q \geq 1.$$

The *principal angles* $\theta_1, \ldots, \theta_q \in [0, \pi/2]$ between F and G are defined recursively by

$$\cos(\theta_k) = \max_{u \in F} \max_{v \in G} u^T v = u_k^T v_k$$

subject to:

$$\begin{array}{ll} \| u \| = \| v \| = 1 & \\ u^T u_i = 0 & i = 1{:}k-1 \\ v^T v_i = 0 & i = 1{:}k-1. \end{array}$$

Note that the principal angles satisfy $0 \leq \theta_1 \leq \cdots \leq \theta_q \leq \pi/2$. The vectors $\{u_1, \ldots, u_q\}$ and $\{v_1, \ldots, v_q\}$ are called the *principal vectors* between the subspaces F and G.

Principal angles and vectors arise in many important statistical applications. The largest principal angle is related to the notion of distance between equidimensional subspaces that we discussed in §2.6.3 If $p = q$ then $\text{dist}(F, G) = \sqrt{1 - \cos(\theta_p)^2} = \sin(\theta_p)$.

If the columns of $Q_F \in \mathbb{R}^{m \times p}$ and $Q_G \in \mathbb{R}^{m \times q}$ define orthonormal bases for F and G respectively, then

$$\max_{\substack{u \in F \\ \|u\|_2 = 1}} \max_{\substack{v \in G \\ \|v\|_2 = 1}} u^T v = \max_{\substack{y \in \mathbb{R}^p \\ \|y\|_2 = 1}} \max_{\substack{z \in \mathbb{R}^q \\ \|z\|_2 = 1}} y^T(Q_F^T Q_G)z$$

From the minimax characterization of singular values given in Theorem 8.6.1 it follows that if $Y^T(Q_F^TQ_G)Z = \text{diag}(\sigma_1, \ldots, \sigma_q)$ is the SVD of $Q_F^TQ_G$, then we may define the u_k, v_k, and θ_k by

$$\begin{aligned} [\, u_1, \ldots, u_p \,] &= Q_F Y \\ [\, v_1, \ldots, v_q \,] &= Q_G Z \\ \cos(\theta_k) &= \sigma_k \qquad k = 1{:}q \end{aligned}$$

Typically, the spaces F and G are defined as the ranges of given matrices $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{m\times q}$. In this case the desired orthonormal bases can be obtained by computing the QR factorizations of these two matrices.

Algorithm 12.4.3 Given $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{m\times q}$ $(p \geq q)$ each with linearly independent columns, the following algorithm computes the orthogonal matrices $U = [\, u_1, \ldots, u_q \,]$ and $V = [\, v_1, \ldots, v_q \,]$ and $\cos(\theta_1), \ldots \cos(\theta_q)$ such that the θ_k are the principal angles between ran(A) and ran(B) and u_k and v_k are the associated principal vectors.

Use Algorithm 5.2.1 to compute the QR factorizations

$$\begin{array}{lll} A = Q_A R_A & Q_A^T Q_A = I_p, & R_A \in \mathbb{R}^{p\times p} \\ B = Q_B R_B & Q_B^T Q_B = I_q, & R_B \in \mathbb{R}^{q\times q} \end{array}$$

$C = Q_A^T Q_B$
Compute the SVD $Y^T C Z = \text{diag}(\cos(\theta_k))$.
$Q_A Y(:, 1{:}q) = [\, u_1, \ldots, u_q \,]$
$Q_B Z = [\, v_1, \ldots, v_q \,]$

This algorithm requires about $4m(q^2 + 2p^2) + 2pq(m + q) + 12q^3$ flops.

The idea of using the SVD to compute the principal angles and vectors is due to Björck and Golub (1973). The problem of rank deficiency in A and B is also treated in this paper.

12.4.4 Intersection of Subspaces

Algorithm 12.4.3 can also be used to compute an orthonormal basis for $\text{ran}(A) \cap \text{ran}(B)$ where $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{m\times q}$

Theorem 12.4.2 *Let* $\{\cos(\theta_k), u_k, v_k\}_{k=1}^q$ *be defined by Algorithm 12.4.3. If the index* s *is defined by* $1 = \cos(\theta_1) = \cdots = \cos(\theta_s) > \cos(\theta_{s+1})$, *then we have*

$$\text{ran}(A) \cap \text{ran}(B) = \text{span}\{u_1, \ldots, u_s\} = \text{span}\{v_1, \ldots, v_s\}\ .$$

Proof. The proof follows from the observation that if $\cos(\theta_k) = 1$, then $u_k = v_k$. □

With inexact arithmetic, it is necessary to compute the approximate multiplicity of the unit cosines in Algorithm 12.4.3.

Example 12.4.3 If

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & 5 \\ 3 & 7 \\ 5 & -1 \end{bmatrix}$$

then the cosines of the principal angles between $\text{ran}(A)$ and $\text{ran}(B)$ are 1.000 and .856.

Problems

P12.4.1 Show that if A and B are m-by-p matrices, with $p \le m$, then

$$\min_{Q^TQ=I_p} \| A - BQ \|_F^2 = \sum_{i=1}^{p} (\sigma_i(A)^2 - 2\sigma_i(B^TA) + \sigma_i(B)^2).$$

P12.4.2 Extend Algorithm 12.4.2 so that it can compute an orthonormal basis for $\text{null}(A_1) \cap \cdots \cap \text{null}(A_s)$.

P12.4.3 Extend Algorithm 12.4.3 to handle the case when A and B are rank deficient.

P12.4.4 Relate the principal angles and vectors between $\text{ran}(A)$ and $\text{ran}(B)$ to the eigenvalues and eigenvectors of the generalized eigenvalue problem

$$\begin{bmatrix} 0 & A^TB \\ B^TA & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \sigma \begin{bmatrix} A^TA & 0 \\ 0 & B^TB \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix}.$$

P12.4.5 Suppose $A, B \in \mathbb{R}^{m \times n}$ and that A has full column rank. Show how to compute a symmetric matrix $X \in \mathbb{R}^{n \times n}$ that minimizes $\| AX - B \|_F$. Hint: Compute the SVD of A.

Notes and References for Sec. 12.4

The problem of minimizing $\| A - BQ \|_F$ over all orthogonal matrices arises in psychometrics. See

B. Green (1952). "The Orthogonal Approximation of an Oblique Structure in Factor Analysis," *Psychometrika 17*, 429–40.

P. Schonemann (1966). "A Generalized Solution of the Orthogonal Procrustes Problem," *Psychometrika 31*, 1–10.

I.Y. Bar-Itzhack (1975). "Iterative Optimal Orthogonalization of the Strapdown Matrix," *IEEE Trans. Aerospace and Electronic Systems 11*, 30–37.

R.J. Hanson and M.J. Norris (1981). "Analysis of Measurements Based on the Singular Value Decomposition," *SIAM J. Sci. and Stat. Comp. 2*, 363–374.

H. Park (1991). "A Parallel Algorithm for the Unbalanced Orthogonal Procrustes Problem," *Parallel Computing 17*, 913–923.

When $B = I$, this problem amounts to finding the closest orthogonal matrix to A. This is equivalent to the polar decomposition problem of §4.2.10. See

A. Björck and C. Bowie (1971). "An Iterative Algorithm for Computing the Best Estimate of an Orthogonal Matrix," *SIAM J. Num. Anal. 8,* 358–64.

N.J. Higham (1986). "Computing the Polar Decomposition—with Applications," *SIAM J. Sci. and Stat. Comp. 7,* 1160–1174.

If A is reasonably close to being orthogonal itself, then Björck and Bowie's technique is more efficient than the SVD algorithm.

The problem of minimizing $\| AX - B \|_F$ subject to the constraint that X is symmetric is studied in

N.J. Higham (1988). "The Symmetric Procrustes Problem," *BIT 28,* 133–43.

Using the SVD to solve the canonical correlation problem is discussed in

A. Björck and G.H. Golub (1973). "Numerical Methods for Computing Angles Between Linear Subspaces," *Math. Comp. 27,* 579–94.

G.H. Golub and H. Zha (1994). "Perturbation Analysis of the Canonical Correlations of Matrix Pairs," *Lin. Alg. and Its Applic. 210,* 3–28.

The SVD has other roles to play in statistical computation.

S.J. Hammarling (1985). "The Singular Value Decomposition in Multivariate Statistics," *ACM SIGNUM Newsletter 20,* 2-25.

12.5 Updating Matrix Factorizations

In many applications it is necessary to re-factor a given matrix $A \in \mathbb{R}^{m \times n}$ after it has been altered in some minimal sense. For example, given that we have the QR factorization of A, we may need to calculate the QR factorization of a matrix A that is obtained by (a) adding a general rank-one matrix to A, (b) appending a row (or column) to A, or (c) deleting a row (or column) from A. In this section we show that in situations like these, it is much more efficient to "update" A's QR factorization than to generate it from scratch. We also show how to update the null space of a matrix after it has been augmented with an additional row.

Before beginning, we mention that there are also techniques for updating the factorizations $PA = LU$, $A = GG^T$, and $A = LDL^T$. Updating these factorizations, however, can be quite delicate because of pivoting requirements and because when we tamper with a positive definite matrix the result may not be positive definite. See Gill, Golub, Murray, and Saunders (1974) and Stewart (1979). Along these lines we briefly discuss hyperbolic transformations and their use in the Cholesky downdating problem.

Familiarity with §3.5, §4.1, §5.1, §5.2, §5.4, and §5.5 is required. Complementary reading includes Gill, Murray, and Wright (1991).

12.5.1 Rank-One Changes

Suppose we have the QR factorization $QR = B \in \mathbb{R}^{m\times n}$ and that we need to compute the QR factorization $B + uv^T = Q_1R_1$ where $u, v \in \mathbb{R}^n$ are given. Observe that

$$B + uv^T = Q(R + wv^T) \qquad (12.5.1)$$

where $w = Q^Tu$. Suppose that we compute rotations $J_{n-1}, \ldots, J_2, J_1$ such that

$$J_1^T \cdots J_{n-1}^T w = \pm\| w \|_2 e_1 .$$

Here, each J_k is a rotation in planes k and $k+1$. (For details, see Algorithm 5.1.3.) If these same Givens rotations are applied to R, it can be shown that

$$H = J_1^T \cdots J_{n-1}^T R \qquad (12.5.2)$$

is upper Hessenberg. For example, in the $n = 4$ case we start with

$$R = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \qquad w = \begin{bmatrix} \times \\ \times \\ \times \\ \times \end{bmatrix}$$

and then update as follows:

$$R = J_3^TR = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \qquad w = J_3^Tw = \begin{bmatrix} \times \\ \times \\ \times \\ 0 \end{bmatrix}$$

$$R = J_2^TR = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \qquad w = J_2^Tw = \begin{bmatrix} \times \\ \times \\ 0 \\ 0 \end{bmatrix}$$

$$H = J_1^TR = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \qquad w = J_1^Tw = \begin{bmatrix} \times \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Consequently,

$$(J_1^T \cdots J_{n-1}^T)(R + wv^T) = H \pm \| w \|_2 e_1 v^T = H_1 \qquad (12.5.3)$$

is also upper Hessenberg.

In Algorithm 5.2.3, we show how to compute the QR factorization of an upper Hessenberg matrix in $O(n^2)$ flops. In particular, we can find Givens rotations G_k , $k = 1{:}n-1$ such that

$$G_{n-1}^T \cdots G_1^T H_1 = R_1 \qquad (12.5.4)$$

is upper triangular. Combining (12.5.1) through (12.5.4) we obtain the QR factorization $B + uv^T = Q_1 R_1$ where

$$Q_1 = QJ_{n-1} \cdots J_1 G_1 \cdots G_{n-1}.$$

A careful assessment of the work reveals that about $26n^2$ flops are required. The vector $w = Q^T u$ requires $2n^2$ flops. Computing H and accumulating the J_k into Q. involves $12n^2$ flops. Finally, computing R_1 and multiplying the G_k into Q involves $12n^2$ flops.

The technique readily extends to the case when B is rectangular. It can also be generalized to compute the QR factorization of $B + UV^T$ where $\text{rank}(UV^T) = p > 1$.

12.5.2 Appending or Deleting a Column

Assume that we have the QR factorization

$$QR = A = [\, a_1, \ldots, a_n \,] \qquad a_i \in \mathbb{R}^m \tag{12.5.5}$$

and partition the upper triangular matrix $R \in \mathbb{R}^{m \times n}$ as follows:

$$R = \begin{array}{c} \left[\begin{array}{ccc} R_{11} & v & R_{13} \\ 0 & r_{kk} & w^T \\ 0 & 0 & R_{33} \end{array} \right] \begin{array}{c} k-1 \\ 1 \\ m-k \end{array} \\ \begin{array}{ccc} k-1 & 1 & n-k \end{array} \end{array} .$$

Now suppose that we want to compute the QR factorization of

$$\tilde{A} = [\, a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_n \,] \in \mathbb{R}^{m \times (n-1)} .$$

Note that $\tilde{A}$ is just A with its kth column deleted and that

$$Q^T \tilde{A} = \left[\begin{array}{cc} R_{11} & R_{13} \\ 0 & w^T \\ 0 & R_{33} \end{array} \right] = H$$

is upper Hessenberg, e.g.,

$$H = \left[\begin{array}{ccccc} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \qquad m = 7,\ n = 6,\ k = 3$$

Clearly, the unwanted subdiagonal elements $h_{k+1,k}, \ldots, h_{n,n-1}$ can be zeroed by a sequence of Givens rotations: $G_{n-1}^T \cdots G_k^T H = R_1$. Here, G_i is

a rotation in planes i and $i+1$ for $i = k{:}n-1$. Thus, if $Q_1 = QG_k \cdots G_{n-1}$ then $\tilde{A} = Q_1 R_1$ is the QR factorization of $\tilde{A}$.

The above update procedure can be executed in $O(n^2)$ flops and is very useful in certain least squares problems. For example, one may wish to examine the significance of the kth factor in the underlying model by deleting the kth column of the corresponding data matrix and solving the resulting LS problem.

In a similar vein, it is useful to be able to compute efficiently the solution to the LS problem after a column has been appended to A. Suppose we have the QR factorization (12.5.5) and now wish to compute the QR factorization of

$$\tilde{A} = [\, a_1, \ldots, a_k, z, a_{k+1}, \ldots, a_n \,]$$

where $z \in \mathbb{R}^m$ is given. Note that if $w = Q^T z$ then

$$Q^T A = \left[\, Q^T a_1, \ldots, Q^T a_k, w, Q^T a_{k+1}, \ldots, Q^T a_n \,\right] = \tilde{A}$$

is upper triangular except for the presence of a "spike" in its $k+1$-st column, e.g.,

$$\tilde{A} = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & 0 & \times \\ 0 & 0 & 0 & \times & 0 & 0 \\ 0 & 0 & 0 & \times & 0 & 0 \end{bmatrix} \qquad m = 7,\ n = 5,\ k = 3$$

It is possible to determine Givens rotations $J_{m-1}, \ldots, J_{k+1}$ so that

$$J_{k+1}^T \cdots J_{m-1}^T w = \begin{bmatrix} w_1 \\ \vdots \\ w_{k+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

with $J_{k+1}^T \cdots J_{m-1}^T \tilde{A} = \tilde{R}$ upper triangular. We illustrate this by continuing with the above example:

$$H = J_6^T \tilde{A} = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & 0 & \times \\ 0 & 0 & 0 & \times & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H = J_5^T H = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & 0 & \times \\ 0 & 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H = J_4^T H = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This update requires $O(mn)$ flops.

12.5.3 Appending or Deleting a Row

Suppose we have the QR factorization $QR = A \in \mathbb{R}^{m \times n}$ and now wish to obtain the QR factorization of

$$\tilde{A} = \begin{bmatrix} w^T \\ A \end{bmatrix}$$

where $w \in \mathbb{R}^n$. Note that

$$\text{diag}(1, Q^T)\tilde{A} = \begin{bmatrix} w^T \\ R \end{bmatrix} = H$$

is upper Hessenberg. Thus, Givens rotations $J_1, \ldots, J_n$ could be determined so $J_n^T \cdots J_1^T H = R_1$ is upper triangular. It follows that

$$\tilde{A} = Q_1 R_1$$

is the desired QR factorization, where $Q_1 = \text{diag}(1, Q) J_1 \cdots J_n$.

No essential complications result if the new row is added between rows k and $k+1$ of A. We merely apply the above with A replaced by PA and Q replaced by PQ where

$$P = \begin{bmatrix} 0 & I_{m-k} \\ I_k & 0 \end{bmatrix}.$$

Upon completion $\text{diag}(1, P^T)Q_1$ is the desired orthogonal factor.

Lastly, we consider how to update the QR factorization $QR = A \in \mathbb{R}^{m\times n}$ when the first row of A is deleted. In particular, we wish to compute the QR factorization of the submatrix A_1 in

$$A = \begin{bmatrix} z^T \\ A_1 \end{bmatrix} \begin{matrix} 1 \\ m-1 \end{matrix}$$

(The procedure is similar when an arbitrary row is deleted.) Let q^T be the first row of Q and compute Givens rotations $G_1, \ldots, G_{m-1}$ such that

$$G_1^T \cdots G_{m-1}^T q = \alpha e_1,$$

where $\alpha = \pm 1$. Note that

$$H = G_1^T \cdots G_{m-1}^T R = \begin{bmatrix} v^T \\ R_1 \end{bmatrix} \begin{matrix} 1 \\ m-1 \end{matrix}$$

is upper Hessenberg and that

$$QG_{m-1} \cdots G_1 = \begin{bmatrix} \alpha & 0 \\ 0 & Q_1 \end{bmatrix}$$

where $Q_1 \in \mathbb{R}^{(m-1)\times(m-1)}$ is orthogonal. Thus,

$$A = \begin{bmatrix} z^T \\ A_1 \end{bmatrix} = (QG_{m-1} \cdots G_1)(G_1^T \cdots G_{m-1}^T R) = \begin{bmatrix} \alpha & 0 \\ 0 & Q_1 \end{bmatrix} \begin{bmatrix} v^T \\ R_1 \end{bmatrix}$$

from which we conclude that $A_1 = Q_1 R_1$ is the desired QR factorization.

12.5.4 Hyperbolic Transformation Methods

Recall that the "R" in $A = QR$ is the transposed Cholesky factor in $A^T A = GG^T$. Thus, there is a close connection between the QR modifications just discussed and analogous modifications of the Cholesky factorization. We illustrate this with the *Cholesky downdating problem* which corresponds to the removal of an A-row in QR. In the Cholesky downdating problem we have the Cholesky factorization

$$GG^T = A^T A = \begin{bmatrix} z^T \\ A_1 \end{bmatrix}^T \begin{bmatrix} z^T \\ A_1 \end{bmatrix} \tag{12.5.6}$$

where $A \in \mathbb{R}^{m\times n}$ with $m > n$ and $z \in \mathbb{R}^n$. Our task is to find a lower triangular G_1 such that $G_1 G_1^T = A_1^T A_1$. There are several approaches to this interesting and important problem. Simply because it is an opportunity to introduce some new ideas, we present a downdating procedure that relies on *hyperbolic transformations*.

We start with a definition. $H \in \mathbb{R}^{m\times m}$ is *pseudo-orthogonal* with respect to the *signature matrix* $S = \text{diag}(\pm 1) \in \mathbb{R}^{m\times m}$ if $H^TSH = S$. Now from (12.5.6) we have $A^TA = A_1^TA_1 + zz^T = GG^T$ and so

$$A_1^TA_1 \;=\; A^TA - zz^T \;=\; GG^T - zz^T \;=\; [\,G\,z\,]\begin{bmatrix} I_n & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} G^T \\ z^T \end{bmatrix}.$$

Define the signature matrix

$$S \;=\; \begin{bmatrix} I_n & 0 \\ 0 & -1 \end{bmatrix} \tag{12.5.7}$$

and suppose that we can find $H \in \mathbb{R}^{(n+1)\times(n+1)}$ such that $H^TSH = S$ with the property that

$$H\begin{bmatrix} G^T \\ z^T \end{bmatrix} \;=\; \begin{bmatrix} G_1^T \\ 0 \end{bmatrix} \tag{12.5.8}$$

is upper triangular. It follows that

$$A_1^TA_1 \;=\; [\,G\,z\,]H^TSH\begin{bmatrix} G^T \\ z^T \end{bmatrix} \;=\; [G_1\,0\,]\,S\begin{bmatrix} G_1 \\ 0 \end{bmatrix} \;=\; G_1G_1^T$$

is the sought after Cholesky factorization.

We now show how to construct the hyperbolic transformation H in (12.5.8) using *hyperbolic rotations*. A 2-by-2 hyperbolic rotation has the form

$$H \;=\; \begin{bmatrix} \cosh(\theta) & -\sinh(\theta) \\ -\sinh(\theta) & \cosh(\theta) \end{bmatrix} \;=\; \begin{bmatrix} c & -s \\ -s & c \end{bmatrix}.$$

Note that if $H \in \mathbb{R}^{2\times 2}$ is a hyperbolic rotation then $H^TSH = S$ where S = diag(-1,1). Paralleling our Givens rotations developments, let us see how hyperbolic rotations can be used for zeroing. From

$$\begin{bmatrix} c & -s \\ -s & c \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \;=\; \begin{bmatrix} r \\ 0 \end{bmatrix} \qquad c^2 - s^2 = 1$$

we obtain the equation $cx_2 = sx_1$. Note that there is no solution to this equation if $x_1 = x_2 \neq 0$, a clue that hyperbolic rotations are not as numerically solid as their Givens rotation counterparts. If $x_1 \neq x_2$ then it is possible to compute the cosh-sinh pair:

$$\begin{array}{l}
\textbf{if } x_2 = 0 \\
\qquad s = 0;\ c = 1 \\
\textbf{else} \\
\qquad \textbf{if } |x_2| < |x_1| \\
\qquad\qquad \tau = x_2/x_1;\ c = 1/\sqrt{1-\tau^2};\ s = c\tau \\
\qquad \textbf{elseif } |x_1| < |x_2| \\
\qquad\qquad \tau = x_1/x_2;\ s = 1/\sqrt{1-\tau^2};\ c = s\tau \\
\qquad \textbf{end} \\
\textbf{end}
\end{array} \tag{12.5.9}$$

Observe that the norm of the hyperbolic rotation produced by this algorithm gets large as x_1 gets close to x_2.

Now any matrix $H = H(p, n+1, \theta) \in \mathbb{R}^{(n+1)\times(n+1)}$ that is the identity everywhere except $h_{p,p} = h_{n+1,n+1} = \cosh(\theta)$ and $h_{p,n+1} = h_{n+1,p} = -\sinh(\theta)$ satisfies $H^TSH = S$ where S is prescribed in (12.5.7). Using (12.5.9), we attempt to generate hyperbolic rotations $H_k = H(1, k, \theta_k)$ for $k = 2{:}n+1$ so that

$$H_n \cdots H_1 \begin{bmatrix} G^T \\ z^T \end{bmatrix} = \begin{bmatrix} \tilde{G}^T \\ 0 \end{bmatrix}.$$

This turns out to be possible if A has full column rank. Hyperbolic rotation H_k zeros entry $(k+1, k)$. In other words, if A has full column rank, then it can be shown that each call to (12.5.9) results in a cosh-sinh pair. See Alexander, Pan, and Plemmons (1988).

12.5.5 Updating the ULV Decomposition

Suppose $A \in \mathbb{R}^{m\times n}$ is rank deficient and that we have a basis for its null space. If we add a row to A,

$$\tilde{A} = \begin{bmatrix} A \\ z^T \end{bmatrix},$$

then how easy is it to compute a null basis for $\tilde{A}$? When a sequence of such update problems are involved the issue is one of *tracking* the null space. Subspace tracking arises in a number of real-time signal processing applications.

Working with the SVD is awkward in this context because $O(n^3)$ flops are required to recompute the SVD of a matrix that has undergone a unit rank perturbation. On the other hand, Stewart (1993) has shown that the null space updating problem becomes $O(n^2)$ per step if we properly couple the ideas of condition estimation of §3.5.4 and complete orthogonal decomposition. Recall from §5.4.2 that a complete orthogonal decomposition is two-sided and reveals the rank of the underlying matrix,

$$U^TAV = \begin{bmatrix} T_{11} & 0 \\ 0 & 0 \end{bmatrix}, \qquad T_{11} \in \mathbb{R}^{r\times r},\ r = \text{rank}(A).$$

A pair of QR factorizations (one with column pivoting) can be used to compute this. In this case $T_{11} = L$ is lower triangular in exact arithmetic. But with noise and roundoff we instead compute

$$U^T A V = \begin{bmatrix} L & 0 \\ H & E \\ 0 & 0 \end{bmatrix} \qquad (12.5.10)$$

where $L \in \mathbb{R}^{r\times r}$ and $E \in \mathbb{R}^{(n-r)\times(n-r)}$ are lower triangular and H and E are "small" compared to $\sigma_{min}(L)$. In this case we refer to (12.5.10) as a *rank-revealing ULV decomposition.*[1] Note that if

$$V = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \qquad U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$$
$$\quad\;\; r \quad n-r \qquad\qquad\quad\; r \quad m-r$$

then the columns of V_2 define an approximate null space:

$$\| AV_2 \|_2 = \| U_2 E \|_2 \le \| E \|_2.$$

Our goal is to produce a rank-revealing ULV decomposition for the row-appended matrix $\tilde{A}$. To be more specific, our aim is to show how to produce updates of L, E, H, V, and (possibly) the rank in $O(n^2)$ flops.

Note that

$$\begin{bmatrix} U & 0 \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} A \\ z^T \end{bmatrix} V = \begin{bmatrix} L & 0 \\ H & E \\ 0 & 0 \\ w^T & y^T \end{bmatrix}.$$

By permuting the bottom row up "underneath" H and E we see that the challenge is to compute a rank-revealing ULV decomposition of

$$\begin{bmatrix} L & 0 \\ H & E \\ w^T & y^T \end{bmatrix} = \left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & 0 & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline w & w & w & w & y & y & y \end{array}\right] \qquad (12.5.11)$$

in $O(n^2)$ flops. Here and in the sequel, we set $r = 4$ and $n = 7$ to illustrate the main ideas. Bear in mind that the h and e entries are small and that

[1] Dual to this is the URV decomposition in which the rank-revealing form is upper triangular. There are updating situations that sometimes favor the manipulation of this form instead of ULV.

we have deduced that the numerical rank is four. In practice, this involves comparisons with a small tolerance as discussed in §5.5.7.

Using zeroing techniques similar to those presented in §12.5.3, the bottom row can be zeroed with a sequence of row rotations giving

$$\left[\begin{array}{c} \tilde{L} \\ \hline 0 \end{array}\right] = \left[\begin{array}{cccc|ccc} \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 & 0 \\ \hline \times & \times & \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & \times & \times & \times \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

Because this zeroing process intermingles the (presumably large) entries of the bottom row with the entries from each of the other rows, the triangular form typically is *not* rank revealing. However, we can restore the rank-revealing structure with a combination of condition estimation and zero-chasing with rotations. Let us assume that with the added row, the new null space has dimension two.

With a reliable condition estimator we produce a unit 2-norm vector p such that

$$\| p^T \tilde{L} \|_2 \approx \sigma_{min}(\tilde{L}).$$

See §3.5.4. Rotations $\{U_{i,i+1}\}_{i=1}^{6}$ can be found such that

$$U_{67}^T U_{56}^T U_{45}^T U_{34}^T U_{23}^T U_{12}^T p = e_8 = I_8(:,8).$$

The matrix

$$H = U_{67}^T U_{56}^T U_{45}^T U_{34}^T U_{23}^T U_{12}^T \tilde{L}$$

is lower Hessenberg and can be restored to a lower triangular form L_+ by a sequence of column rotations:

$$L_+ = H V_{12} V_{23} V_{34} V_{45} V_{56} V_{67}.$$

It follows that

$$e_8^T L_+ = \left(e_8^T H\right) V_{12} V_{23} V_{34} V_{45} V_{56} V_{67} = \left(p^T \tilde{L}\right) V_{12} V_{23} V_{34} V_{45} V_{56} V_{67}$$

has approximate norm $\sigma_{min}(\tilde{L})$. Thus, we obtain a lower triangular matrix of the form

$$\left[\begin{array}{cccccc|c} \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & \times & \times & 0 \\ \hline h & h & h & h & h & h & e \end{array}\right]$$

with small h's and e. We can repeat the condition estimation and zero chasing on the leading 6-by-6 portion thereby producing (perhaps) another row of small numbers:

$$\left[\begin{array}{ccccc|cc} \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & 0 & 0 \\ \hline h & h & h & h & h & e & 0 \\ h & h & h & h & h & e & e \end{array}\right].$$

(If not, then the revealed rank is 6.) Continuing in this way, we can restore any lower triangular matrix to rank-revealing form.

In the event that the y vector in (12.5.11) is small, we can reach rank-revealing form by a different, more efficient route. We start with a sequence of left and right Givens rotations to zero all but the first component of y:

$$\stackrel{V_{67}}{\longrightarrow} \left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & 0 & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & e \\ h & h & h & h & e & e & e \\ \hline x & x & x & x & y & y & 0 \end{array}\right] \quad \stackrel{U_{67}}{\longrightarrow} \left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & 0 & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline x & x & x & x & y & y & 0 \end{array}\right]$$

$$\stackrel{V_{56}}{\longrightarrow} \left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & 0 & 0 & 0 \\ \hline h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline x & x & x & x & y_* & 0 & 0 \end{array}\right] \quad \stackrel{U_{56}}{\longrightarrow} \left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & 0 & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline x & x & x & x & y_* & 0 & 0 \end{array}\right]$$

Here, "U_{ij}" means a rotation of rows i and j and "V_{ij}" means a rotation of columns i and j. It is important to observe that there is no intermingling of small and large numbers during this process. The h's and e's are still small.

Following this, we produce a sequence of rotations that transform the matrix to

$$\left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & 0 & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline y & y & y & y & y & 0 & 0 \end{array}\right] \qquad (12.5.12)$$

where all the y's are small:

$$\stackrel{U_{48}}{\longrightarrow} \left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & \mu & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & e \\ h & h & h & h & e & e & e \\ \hline x & x & x & 0 & y & 0 & 0 \end{array}\right] \quad \stackrel{U_{38}}{\longrightarrow} \left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & \mu & 0 & 0 \\ \ell & \ell & \ell & \ell & \mu & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline x & x & 0 & 0 & y & 0 & 0 \end{array}\right]$$

$$\stackrel{U_{28}}{\longrightarrow} \left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & \mu & 0 & 0 \\ \ell & \ell & \ell & 0 & \mu & 0 & 0 \\ \ell & \ell & \ell & \ell & \mu & 0 & 0 \\ \hline h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline x & 0 & 0 & 0 & y & 0 & 0 \end{array}\right] \quad \stackrel{U_{18}}{\longrightarrow} \left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & \mu & 0 & 0 \\ \ell & \ell & 0 & 0 & \mu & 0 & 0 \\ \ell & \ell & \ell & 0 & \mu & 0 & 0 \\ \ell & \ell & \ell & \ell & \mu & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline 0 & 0 & 0 & 0 & y_{**} & 0 & 0 \end{array}\right]$$

Note that y_{**} is small because of 2-norm preservation. Column rotations

in planes (1,5), (2,5), (3,5), and (4,5) can remove the μ's:

$$\overset{V_{15}}{\longrightarrow}\left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & \mu & 0 & 0 \\ \ell & \ell & \ell & 0 & \mu & 0 & 0 \\ \ell & \ell & \ell & \ell & \mu & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline y & 0 & 0 & 0 & y & 0 & 0 \end{array}\right] \quad \overset{V_{25}}{\longrightarrow}\left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & \mu & 0 & 0 \\ \ell & \ell & \ell & \ell & \mu & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline y & y & 0 & 0 & y & 0 & 0 \end{array}\right]$$

$$\overset{V_{35}}{\longrightarrow}\left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & \mu & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline y & y & y & 0 & y & 0 & 0 \end{array}\right] \quad \overset{V_{45}}{\longrightarrow}\left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & 0 & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline y & y & y & y & y & 0 & 0 \end{array}\right]$$

thus producing the structure displayed in (12.5.12). All the y's are small and thus a sequence of row rotations $U_{57}, U_{47}, \ldots, U_{17}$, can be constructed to clean out the bottom row giving the rank-revealed form

$$\left[\begin{array}{cccc|ccc} \ell & 0 & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & 0 & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & 0 & 0 & 0 & 0 \\ \ell & \ell & \ell & \ell & 0 & 0 & 0 \\ \hline h & h & h & h & e & 0 & 0 \\ h & h & h & h & e & e & 0 \\ h & h & h & h & e & e & e \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

Problems

P12.5.1 Suppose we have the QR factorization for $A \in \mathbb{R}^{m \times n}$ and now wish to minimize $\| (A + uv^T)x - b \|_2$ where $u, b \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$ are given. Give an algorithm for solving this problem that requires $O(mn)$ flops. Assume that Q must be updated.

P12.5.2 Suppose we have the QR factorization $QR = A \in \mathbb{R}^{m \times n}$. Give an algorithm for computing the QR factorization of the matrix A obtained by deleting the kth row of A. Your algorithm should require $O(mn)$ flops.

P12.5.3 Suppose $T \in \mathbb{R}^{n \times n}$ is tridiagonal and symmetric and that $v \in \mathbb{R}^n$. Show how the Lanczos algorithm can be used (in principle) to compute an orthogonal $Q \in \mathbb{R}^{n \times n}$ in $O(n^2)$ flops such that $Q^T(T + vv^T)Q = \tilde{T}$ is also tridiagonal.

P12.5.4 Suppose

$$A = \begin{bmatrix} c^T \\ B \end{bmatrix} \qquad c \in \mathbb{R}^n, \; B \in \mathbb{R}^{(m-1) \times n}$$

has full column rank and $m > n$. Using the Sherman-Morrison-Woodbury formula show that

$$\frac{1}{\sigma_{min}(B)} \leq \frac{1}{\sigma_{min}(A)} + \frac{\| (A^TA)^{-1}c \|_2^2}{1 - c^T(A^TA)^{-1}c} .$$

P12.5.5 As a function of x_1 and x_2, what is the 2-norm of the hyperbolic rotation produced by (12.5.9)?

P12.5.6 Show that the hyperbolic reduction in §12.5.4 does not breakdown if A has full column rank.

P12.5.7 Assume

$$A = \begin{bmatrix} R & H \\ 0 & E \end{bmatrix}$$

where R and E are square with

$$\rho = \frac{\| E \|_2}{\sigma_{min}(R)} < 1.$$

Show that if

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

is orthogonal and

$$\begin{bmatrix} R & H \\ 0 & E \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} = \begin{bmatrix} R_1 & 0 \\ H_1 & E_1 \end{bmatrix},$$

then $\| H_1 \|_2 \leq \rho \| H \|_2$.

Notes and References for Sec. 12.5

Numerous aspects of the updating problem are presented in

P.E. Gill, G.H. Golub, W. Murray, and M.A. Saunders (1974). "Methods for Modifying Matrix Factorizations," *Math. Comp. 28*, 505-35.

Applications in the area of optimization are covered in

R.H. Bartels (1971). "A Stabilization of the Simplex Method," *Numer. Math. 16*, 414-434.

P.E. Gill, W. Murray, and M.A. Saunders (1975). "Methods for Computing and Modifying the LDV Factors of a Matrix," *Math. Comp. 29*, 1051–77.

D. Goldfarb (1976). "Factored Variable Metric Methods for Unconstrained Optimization," *Math. Comp. 30*, 796–811.

J.E. Dennis and R.B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.

W.W. Hager (1989). "Updating the Inverse of a Matrix," *SIAM Review 31*, 221-239.

S.K. Eldersveld and M.A. Saunders (1992). "A Block-LU Update for Large-Scale Linear Programming," *SIAM J. Matrix Anal. Appl. 13*, 191–201.

Updating issues in the least squares setting are discussed in

J. Daniel, W.B. Gragg, L. Kaufman, and G.W. Stewart (1976). "Reorthogonaization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization," *Math. Comp. 30*, 772-95.

S. Qiao (1988). "Recursive Least Squares Algorithm for Linear Prediction Problems," *SIAM J. Matrix Anal. Appl. 9*, 323 328.

Å. Björck, H. Park, and L. Eldén (1994). "Accurate Downdating of Least Squares Solutions," *SIAM J. Matrix Anal. Appl. 15*, 549–568.

S.J. Olszanskyj, J.M. Lebak, and A.W. Bojanczyk (1994). "Rank-k Modification Methods for Recursive Least Squares Problems," *Numerical Algorithms 7*, 325–354.

L. Eldén and H. Park (1994). "Block Downdating of Least Squares Solutions," *SIAM J. Matrix Anal. Appl. 15*, 1018–1034.

Another important topic concerns the updating of condition estimates:

W.R. Ferng, G.H. Golub, and R.J. Plemmons (1991). "Adaptive Lanczos Methods for Recursive Condition Estimation," *Numerical Algorithms 1*, 1-20.

G. Shroff and C.H. Bischof (1992). "Adaptive Condition Estimation for Rank-One Updates of QR Factorizations," *SIAM J. Matrix Anal. Appl. 13*, 1264 1278.

D.J. Pierce and R.J. Plemmons (1992). "Fast Adaptive Condition Estimation," *SIAM J. Matrix Anal. Appl. 13*, 274–291.

Hyperbolic transformations are discussed in

G.H. Golub (1969). "Matrix Decompositions and Statistical Computation," in *Statistical Computation*, ed., R.C. Milton and J.A. Nelder, Academic Press, New York, pp. 365–97.

C.M. Rader and A.O. Steinhardt (1988). "Hyperbolic Householder Transforms," *SIAM J. Matrix Anal. Appl. 9*, 269–290.

S.T. Alexander, C.T. Pan, and R.J. Plemmons (1988). "Analysis of a Recursive Least Squares Hyperbolic Rotation Algorithm for Signal Processing," *Lin. Alg. and Its Applic. 98*, 3–40.

G. Cybenko and M. Berry (1990). "Hyperbolic Householder Algorithms for Factoring Structured Matrices," *SIAM J. Matrix Anal. Appl. 11*, 499–520.

A.W. Bojanczyk, R. Onn, and A.O. Steinhardt (1993). "Existence of the Hyperbolic Singular Value Decomposition," *Lin. Alg. and Its Applic. 185*, 21–30.

Cholesky update issues have also attracted a lot of attention.

G.W. Stewart (1979). "The Effects of Rounding Error on an Algorithm for Downdating a Cholesky Factorization," *J. Inst. Math. Applic. 23*, 203–13.

A.W. Bojanczyk, R.P. Brent, P. Van Dooren, and F.R. de Hoog (1987). "A Note on Downdating the Cholesky Factorization," *SIAM J. Sci. and Stat. Comp. 8*, 210–221.

C.S. Henkel, M.T. Heath, and R.J. Plemmons (1988). "Cholesky Downdating on a Hypercube," in G. Fox (1988), 1592–1598.

C.-T. Pan (1993). "A Perturbation Analysis of the Problem of Downdating a Cholesky Factorization," *Lin. Alg. and Its Applic. 183*, 103–115.

L. Eldén and H. Park (1994). "Perturbation Analysis for Block Downdating of a Cholesky Decomposition," *Numer. Math. 68*, 457–468.

Updating and downdating the ULV and URV decompositions and related topics are covered in

C.H. Bischof and G.M. Shroff (1992). "On Updating Signal Subspaces," *IEEE Trans. Signal Proc. 40*, 96–105.

G.W. Stewart (1992). "An Updating Algorithm for Subspace Tracking," *IEEE Trans. Signal Proc. 40*, 1535–1541.

G.W. Stewart (1993). "Updating a Rank-Revealing ULV Decomposition," *SIAM J. Matrix Anal. Appl. 14*, 494–499.

G.W. Stewart (1994). "Updating URV Decompositions in Parallel," *Parallel Computing 20*, 151 -172.

H. Park and L. Eldén (1995). "Downdating the Rank-Revealing URV Decomposition," *SIAM J. Matrix Anal. Appl. 16*, 138–155.

Finally, we mention the following paper concerned with SVD updating:

M. Moonen, P. Van Dooren, and J. Vandewalle (1992). "A Singular Value Decomposition Updating Algorithm," *SIAM J. Matrix Anal. Appl. 13*, 1015–1038.

12.6 Modified/Structured Eigenproblems

In this section we treat an array of constrained, inverse, and structured eigenvalue problems. Although the examples are not related, collectively they show how certain special eigenproblems can be solved using the basic factorization ideas presented in earlier chapters.

The dependence of this section upon earlier portions of the book is as follows:

§§5.1, 5.2, 8.1, 8.3	→	§12.6.1
§§8.1, 8.3, 9.1	→	§12.6.2
§§4.7, 8.1	→	§12.6.3
§§5.1, 5.2, 5.4, 7.4, 8.1, 8.2, 8.3, 8.6	→	§12.6.4

12.6.1 A Constrained Eigenvalue Problem

Let $A \in \mathbb{R}^{n\times n}$ be symmetric. The gradient of $r(x) = x^TAx/x^Tx$ is zero if and only if x is an eigenvector of A. Thus the stationary values of $r(x)$ are therefore the eigenvalues of A.

In certain applications it is necessary to find the stationary values of $r(x)$ subject to the constraint $C^Tx = 0$ where $C \in \mathbb{R}^{n\times p}$ with $n \geq p$. Suppose

$$Q^TCZ = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} \qquad r = \text{rank}(C)$$
$$\qquad\qquad r \quad p-r$$

is a complete orthogonal decomposition of C. Define $B \in \mathbb{R}^{n\times n}$ by

$$Q^TAQ = B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix}$$
$$\qquad\qquad r \quad n-r$$

and set

$$y = Q^Tx = \begin{bmatrix} u \\ v \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} .$$

Since $C^Tx = 0$ transforms to $S^Tu = 0$, the original problem becomes one of finding the stationary values of $r(y) = y^TBy/y^Ty$ subject to the constraint that $u = 0$. But this amounts merely to finding the stationary values (eigenvalues) of the $(n-r)$-by-$(n-r)$ symmetric matrix B_{22}.

12.6.2 Two Inverse Eigenvalue Problems

Consider the $r = 1$ case in the previous subsection. Let $\tilde{\lambda}_1 \leq \ldots \leq \tilde{\lambda}_{n-1}$ be the stationary values of x^TAx/x^Tx subject to the constraint $c^Tx = 0$. From Theorem 8.1.7, it is easy to show that these stationary values interlace the eigenvalues λ_i of A:

$$\lambda_n \leq \tilde{\lambda}_{n-1} \leq \lambda_{n-1} \leq \cdots \leq \lambda_2 \leq \tilde{\lambda}_1 \leq \lambda_1.$$

Now suppose that A has distinct eigenvalues and that we are *given* the values $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{n-1}$ that satisfy

$$\lambda_n < \tilde{\lambda}_{n-1} < \lambda_{n-1} < \cdots < \lambda_2 < \tilde{\lambda}_1 < \lambda_1 .$$

We seek to determine a unit vector $c \in \mathbb{R}^n$ such that the $\tilde{\lambda}_i$ are the stationary values of x^TAx subject to $x^Tx = 1$ and $c^Tx = 0$.

In order to determine the properties that c must have, we use the method of Lagrange multipliers. Equating the gradient of

$$\phi(x, \lambda, \mu) = x^TAx - \lambda(x^Tx - 1) + 2\mu x^Tc$$

to zero we obtain the important equation $(A - \lambda I)x = -\mu c$. Thus, $A - \lambda I$ is nonsingular and so $x = -\mu(A - \lambda I)^{-1}c$. Applying c^T to both sides of this equation and substituting the eigenvalue decomposition $Q^TAQ = \text{diag}(\lambda_i)$ we obtain

$$0 = \sum_{i=1}^{n} \frac{d_i^2}{\lambda_i - \lambda}$$

where $d = Q^Tc$, i.e.,

$$p(\lambda) \equiv \sum_{i=1}^{n} d_i^2 \prod_{\substack{j=1 \\ j \neq i}}^{n} (\lambda_j - \lambda) = 0 .$$

Notice that $1 = \| c \|_2^2 = \| d \|_2 = d_1^2 + \cdots + d_n^2$ is the coefficient of $(-\lambda)^{n-1}$. Since $p(\lambda)$ is a polynomial having zeroes $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{n-1}$ we must have

$$p(\lambda) = \prod_{j=1}^{n-1} (\tilde{\lambda}_j - \lambda) .$$

It follows from these two formulas for $p(\lambda)$ that

$$d_k^2 = \frac{\displaystyle\prod_{j=1}^{n-1} (\tilde{\lambda}_j - \lambda_k)}{\displaystyle\prod_{\substack{j=1 \\ j \neq k}}^{n-1} (\lambda_j - \lambda_k)} \qquad k = 1{:}n . \qquad (12.6.1)$$

This determines each d_k up to its sign. Thus there are 2^n different solutions $c = Qd$ to the original problem.

A related inverse eigenvalue problem involves finding a tridiagonal matrix

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & & \beta_{n-1} & \alpha_n \end{bmatrix}$$

such that T has prescribed eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ and $T(2{:}n, 2{:}n)$ has prescribed eigenvalues $\{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{n-1}\}$ with

$$\lambda_1 > \tilde{\lambda}_1 > \lambda_2 > \cdots > \lambda_{n-1} > \tilde{\lambda}_{n-1} > \lambda_n \,.$$

We show how to compute the tridiagonal T via the Lanczos process. Note that the $\tilde{\lambda}_i$ are the stationary values of

$$\phi(y) = \frac{y^T \Lambda y}{y^T y}$$

subject to $d^T y = 0$ where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and d is specified by (12.6.1). If we apply the Lanczos iteration (9.1.3) with $A = \Lambda$ and $q_1 = d$, then it produces an orthogonal matrix Q and a tridiagonal matrix T such that $Q^T \Lambda Q = T$. With the definition $x = Q^T y$, it follows that the $\tilde{\lambda}_i$ are the stationary values of

$$\psi(x) = \frac{x^T T x}{x^T x}$$

subject to $e_1^T x = 0$. But these are precisely the eigenvalues of $T(2{:}n, 2{:}n)$!

12.6.3 A Toeplitz Eigenproblem

Assume that

$$T = \begin{bmatrix} 1 & r^T \\ r & G \end{bmatrix}$$

is symmetric, positive definite, and Toeplitz with $r \in \mathbb{R}^{n-1}$. Our goal is to compute the smallest eigenvalue $\lambda_{min}(T)$ of T given that

$$\lambda_{min}(T) < \lambda_{min}(G).$$

This problem is considered in Cybenko and Van Loan (1986) and has applications in signal processing.

Suppose

$$\begin{bmatrix} 1 & r^T \\ r & G \end{bmatrix} \begin{bmatrix} \alpha \\ y \end{bmatrix} = \lambda \begin{bmatrix} \alpha \\ y \end{bmatrix},$$

i.e.,

$$\begin{aligned} \alpha + r^T y &= \lambda\alpha \\ \alpha r + G y &= \lambda y. \end{aligned}$$

If $\lambda \notin \lambda(G)$, then $y = -\alpha(G - \lambda I)^{-1} r$, $\alpha \neq 0$, and

$$\alpha + r^T \left[-\alpha(G - \lambda I)^{-1} r \right] = \lambda\alpha.$$

Thus, λ is a zero of the rational function

$$f(\lambda) = 1 - \lambda - r^T (G - \lambda I)^{-1} r.$$

We have dealt with similar functions in §8.5 and §12.1. In this case, f always has a negative slope

$$f'(\lambda) \;=\; -1 - \| \, (G - \lambda I)^{-1} r \, \|_2^2 \leq -1.$$

If $\lambda < \lambda_{min}(G)$, then it also has a negative second derivative:

$$f''(\lambda) \;=\; -2r^T (G - \lambda I)^{-3} r \leq 0.$$

Using these facts it can be shown that if

$$\lambda_{min}(T) \leq \lambda^{(0)} < \lambda_{min}(G), \tag{12.6.2}$$

then the Newton iteration

$$\lambda^{(k+1)} \;=\; \lambda^{(k)} - \frac{f(\lambda^{(k)})}{f'(\lambda^{(k)})} \tag{12.6.3}$$

converges to $\lambda_{min}(T)$ monotonically from the right. Note that

$$\lambda^{(k+1)} = \lambda^{(k)} + \frac{1 + r^T w - \lambda^{(k)}}{1 + w^T w}$$

where w solves the "shifted" Yule-Walker system

$$(G - \lambda^{(k)} I) w = -r.$$

Since, $\lambda^{(k)} < \lambda_{min}(G)$, this system is positive definite and Algorithm 4.7.1 is applicable if we simply apply it to the normalized Toeplitz matrix $(G - \lambda^{(k)} I)/(1 - \lambda^{(k)})$.

A starting value that satisfies (12.6.2) can be obtained by examining the Durbin algorithm when it is applied to $T_\lambda = (T - \lambda I)/(1 - \lambda)$. For this matrix the "$r$" vector is $r/(1 - \lambda)$ and so the Durbin algorithm (4.7.1) transforms to

$r = r/(1 - \lambda)$
$y^{(1)} = -r_1$
for $k = 1{:}n-1$
  $\beta_k = 1 + [r^{(k)}]^T y^{(k)}$ (12.6.4)
  $\alpha_k = -(r_{k+1} + r^{(k)^T} E_k y^{(k)})/\beta_k$
  $z^{(k)} = y^{(k)} + \alpha_k E_k y^{(k)}$
  $y^{(k+1)} = \begin{bmatrix} z^{(k)} \\ \alpha_k \end{bmatrix}$
end

From the discussion in §4.7.2 we know that $\beta_1, \ldots, \beta_k > 0$ implies that $T_\lambda(1{:}k+1, 1{:}k+1)$ is positive definite. Hence, a suitably modified (12.6.4) can be used to compute $m(\lambda)$, the largest index m such that $\beta_1, \ldots, \beta_m$ are all positive but that $\beta_{m+1} \le 0$. Note that if $m(\lambda) = n-2$, then (12.6.2) holds. This suggests the following bisection scheme:

Choose L and R so $L \le \lambda_{min}(T) < \lambda_{min}(G) \le R$.
Until $m = n - 2$
  $\lambda = (L+R)/2$
  $m = m(\lambda)$
  if $m < n-2$ (12.6.5)
    $R = \lambda$
  end
  if $m = n-1$
    $L = \lambda$
  end
end

The bracketing interval $[L, R]$ always contains a λ such that $m(\lambda) = n-2$ and so the current λ has this property upon termination.

There are several possible choices for a starting interval. One idea is to set $L = 0$ and $R = 1 - |r_1|$ since

$$0 < \lambda_{min}(T) < \lambda_{min}(G) \le \lambda_{min}\left(\begin{bmatrix} 1 & r_1 \\ r_1 & 1 \end{bmatrix}\right) = 1 - |r_1|$$

where the upper bound follows from Theorem 8.1.7.

Note that the iterations in (12.6.4) and (12.6.5) involve at most $O(n^2)$ flops. A heuristic argument that $O(\log n)$ iterations are required is given in Cybenko and Van Loan (1986).

12.6.4 An Orthogonal Matrix Eigenproblem

Computing the eigenvalues and eigenvectors of a real orthogonal matrix $A \in \mathbb{R}^{n \times n}$ is a problem that arises in signal processing, see Cybenko (1985).

The eigenvalues of A are on the unit circle and moreover,

$$\cos(\theta) \pm i\sin(\theta) \in \lambda(A) \quad \Leftrightarrow \quad \cos(\theta) \in \lambda\left(\frac{A+A^{-1}}{2}\right) = \lambda\left(\frac{A+A^T}{2}\right).$$

This suggests computing $\mathrm{Re}(\lambda(A))$ via the Schur decomposition

$$Q^T\left(\frac{A+A^T}{2}\right)Q = \mathrm{diag}(\cos(\theta_1),\ldots,\cos(\theta_n))$$

and then computing $\mathrm{Im}(\lambda(A))$ with the formula $s = \sqrt{1-c^2}$. Unfortunately, if $|c| \approx 1$, then this formula does not produce an accurate sine because of floating point cancellation. We could work with the skew-symmetric matrix $(A - A^T)/2$ to get the "small sine" eigenvalues, but then we are talking about a method that requires a pair of full Schur decomposition problems and the approach begins to lose its appeal.

A way around these difficulties that involves an interesting SVD application is proposed by Ammar, Gragg, and Reichel (1986). We present just the eigenvalue portion of their algorithm. The derivation is instructive because it involves practically every decomposition that we have studied.

The first step is to orthogonally reduce A to upper Hessenberg form, $Q^TAQ = H$. (Frequently, A is already in Hessenberg form.) Without loss of generality, we may assume that H is unreduced with positive subdiagonal elements.

If n is odd, then it must have a real eigenvalue because the eigenvalues of a real matrix come in complex conjugate pairs. In this case it is possible to deflate the problem with $O(n)$ work to size $n-1$ by carefully working with the eigenvector equation $Hx = x$ (or $Hx = -x$). See Gragg (1986). Thus, we may assume that n is even.

For $1 \le k \le n-1$, define the reflection $G_k \in \mathbb{R}^{n\times n}$ by

$$G_k = G_k(\phi_k) = \begin{bmatrix} I_{k-1} & 0 & 0 & 0 \\ 0 & -c_k & s_k & 0 \\ 0 & s_k & c_k & 0 \\ 0 & 0 & 0 & I_{n-k-1} \end{bmatrix}$$

where $c_k = \cos(\phi_k)$, $s_k = \sin(\phi_k)$, and $0 < \phi_k < \pi$. It is possible to determine $G_1,\ldots,G_{n-1}$ such that

$$H = (G_1 \cdots G_{n-1})\,\mathrm{diag}(1,\ldots,1,-c_n)$$

where $c_n = \pm 1$. This is just the QR decomposition of H. The sines $s_1,\ldots,s_{n-1}$ are the subdiagonal entries of H. The "R" matrix is diagonal because it is orthogonal and triangular. Since the determinant of a reflection is -1, $\det(H) = c_n$. This quantity is the product of H's eigenvalues and so if $c_n = -1$, then $\{-1,1\} \subseteq \lambda(H)$. In this situation it is also possible to deflate.

So altogether we may assume that n is even and

$$H = G_1(\phi_1)\cdots G_{n-1}(\phi_{n-1})G_n(\phi_n)$$

where $G_n = G_n(\phi_n) = \text{diag}(1,\ldots,1,-c_n)$ and $c_n = 1$. Designate the sought after eigenvalues by

$$\lambda(H) = \{\, cos(\theta_k) \pm i\cdot\sin(\theta_k)\,\}_{k=1}^{m} \tag{12.6.4}$$

where $m = n/2$.

The cosines $c_1,\ldots,c_n$ are called the *Schur parameters* and as we mentioned, the corresponding sines are the subdiagonal entries of H. Using these numbers it is possible to construct *explicitly* a pair of bidiagonal matrices $B_C, B_S \in \mathbb{R}^{n\times n}$ with the property that

$$\sigma(B_C(1{:}m,1{:}m)) = \{\cos(\theta_1/2),\ldots,\cos(\theta_m/2)\} \tag{12.6.5}$$
$$\sigma(B_S(1{:}m,1{:}m)) = \{\sin(\theta_1/2),\ldots,\sin(\theta_m/2)\} \tag{12.6.6}$$

The singular values of $B_C(1{:}m,1{:}m)$ and $B_S(1{:}m,1{:}m)$ can be computed using the bidiagonal SVD algorithm. The angle θ_k can be accurately computed from $\sin(\theta_k/2)$ if $0 < \theta_k \le \pi/2$ and accurately computed from $\cos(\theta_k/2)$ if $\pi/2 \le \theta_k < \pi$. The construction of B_C and B_S is based on three facts:

1. H is similar to
$$\tilde{H} = H_oH_e$$
where H_o and H_e are the odd and even reflection products
$$H_o = G_1G_3\cdots G_{n-1}$$
$$H_e = G_2G_4\cdots G_n.$$
These matrices are block diagonal with 2-by-2 and 1-by-1 blocks, i.e.,
$$H_o = \text{diag}(R(\phi_1), R(\phi_3),\ldots,R(\phi_{n-1})) \tag{12.6.7}$$
$$H_e = \text{diag}(1, R(\phi_2), R(\phi_4),\ldots,R(\phi_{n-2}),-1) \tag{12.6.8}$$
where
$$R(\phi) = \begin{bmatrix} -\cos(\phi) & \sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{12.6.9}$$

2. The eigenvalues of the symmetric tridiagonal matrices
$$C = \frac{H_o + H_e}{2} \quad \text{and} \quad S = \frac{H_o - H_e}{2} \tag{12.6.10}$$
are given by
$$\lambda(C) = \{\,\pm\cos(\theta_1/2),\ldots,\pm\cos(\theta_m/2)\,\} \tag{12.6.11}$$
$$\lambda(S) = \{\,\pm\sin(\theta_1/2),\ldots,\pm\sin(\theta_m/2)\,\}. \tag{12.6.12}$$

3. It is possible to construct bidiagonalizations

$$U_C^T C V_C = B_C \qquad \text{and} \qquad U_S^T S V_S = B_S$$

that satisfy (12.6.5) and (12.6.6). The transformations U_C, V_C, U_S, and V_S are products of known reflections G_k and simple permutations.

We begin the verification of these three facts by showing that H is similar to H_oH_e. The $n = 8$ case is sufficient for this purpose. Define the orthogonal matrix P by

$$P = F_7F_5F_3 \qquad \text{where} \qquad \begin{cases} F_3 = G_3G_4G_5G_6G_7G_8 \\ F_5 = G_5G_6G_7G_8 \\ F_7 = G_7G_8. \end{cases}$$

Since reflections are symmetric and $G_iG_j = G_jG_i$ if $|i-j| \geq 2$, we see that

$$\begin{aligned} F_3HF_3^T &= (G_3G_4G_5G_6G_7G_8)(G_1G_2G_3G_4G_5G_6G_7G_8)(G_3G_4G_5G_6G_7G_8)^T \\ &= (G_3G_4G_5G_6G_7G_8)G_1G_2 \\ &= G_1G_3G_2G_4G_5G_6G_7G_8, \end{aligned}$$

$$\begin{aligned} F_5(F_3HF_3^T)F_5^T &= (G_5G_6G_7G_8)(G_1G_3G_2G_4G_5G_6G_7G_8)(G_5G_6G_7G_8)^T, \\ &= (G_5G_6G_7G_8)G_1G_3G_2G_4 \\ &= G_1G_3G_5G_2G_4G_6G_7G_8 \end{aligned}$$

$$\begin{aligned} PHP^T &= F_7(F_5F_3HF_3^TF_5^T)F_7^T \\ &= (G_7G_8)(G_1G_3G_5G_2G_4G_6G_7G_8)(G_7G_8)^T \\ &= (G_1G_3G_5G_7)(G_2G_4G_6G_8) \equiv H_oH_e. \end{aligned}$$

The second of the three facts that we need to establish relates the eigenvalues of $\tilde{H} = H_0H_e$ to the eigenvalues of the C and S matrices defined in (12.6.10). It follows from (12.6.7) and (12.6.8) that these matrices are symmetric, tridiagonal, and unreduced, e.g.,

$$C = \frac{1}{2}\begin{bmatrix} -c_1 & s_1 & 0 & 0 \\ s_1 & c_1 - c_2 & s_2 & 0 \\ 0 & s_2 & c_2 - c_3 & s_3 \\ 0 & 0 & s_3 & c_3 - c_4 \end{bmatrix}$$

$$S = \frac{1}{2}\begin{bmatrix} -c_1 & s_1 & 0 & 0 \\ s_1 & c_1 + c_2 & -s_2 & 0 \\ 0 & -s_2 & -c_2 - c_3 & s_3 \\ 0 & 0 & s_3 & c_3 + c_4 \end{bmatrix}.$$

By working with the definitions it is easy to verify that

$$\frac{\tilde{H} + \tilde{H}^T}{2} = \frac{H_oH_e + (H_oH_e)^{-1}}{2} = \frac{H_oH_e + H_eH_o}{2} = 2C^2 - I$$

and

$$\frac{\tilde{H} + \tilde{H}^T}{2i} = \frac{H_oH_e - (H_oH_e)^{-1}}{2i} = \frac{H_oH_e - H_eH_o}{2i} = -2iSC.$$

This shows that $\mathrm{Re}(\lambda(\tilde{H})) = \lambda(2C^2 - I)$ and $\mathrm{Im}(\lambda(\tilde{H})) = \lambda(-2iCS)$ thereby establishing (12.6.11) and (12.6.12).

Instead of thinking of these half-angle cosines and sines as eigenvalues of n-by-n matrices, it is more efficient to think of them as singular values of m-by-m matrices. This brings us to the bidiagonalization of C and S. The orthogonal equivalence transformations that carry out this task are based upon the Schur decompositions of H_o and H_e. A 2-by-2 reflection $R(\phi)$ defined by (12.6.9) has eigenvalues 1 and -1 and the following Schur decomposition:

$$R(\phi/2)R(\phi)R(\phi/2) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Thus, if

$$\begin{aligned} Q_o &= \mathrm{diag}(R(\phi_1/2), R(\phi_3/2), \ldots, R(\phi_{n-1}/2)) \\ Q_e &= \mathrm{diag}(1, R(\phi_2/2), R(\phi_4/2), \ldots, R(\phi_{n-2}/2), -1) \end{aligned}$$

then from (12.6.7) and (12.6.8) H_o and H_e have the following Schur decompositions:

$$\begin{aligned} Q_oH_oQ_o &= D_o = \mathrm{diag}(1, -1, 1, -1, \cdots, 1, -1) \\ Q_eH_eQ_e &= D_e = \mathrm{diag}(1, 1, -1, 1, -1, \cdots, 1, -1, -1). \end{aligned}$$

The matrices

$$\begin{aligned} C^{(1)} &= Q_oCQ_e = \frac{1}{2}Q_o(H_o + H_e)Q_e = \frac{1}{2}(D_o(Q_oQ_e) + (Q_oQ_e)D_e) \\ S^{(1)} &= Q_oSQ_e = \frac{1}{2}Q_o(H_o - H_e)Q_e = \frac{1}{2}(D_o(Q_oQ_e) - (Q_oQ_e)D_e) \end{aligned}$$

have the same singular values as C and S respectively. To analyze their structure we first note that Q_0Q_e is banded:

$$Q_oQ_e = \begin{bmatrix} \times & \times & \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times & \times & 0 \\ 0 & 0 & 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times & \times & \times \end{bmatrix}.$$

(The main ideas from this point on are amply communicated with $n = 8$ examples.) If $D_o(i,i)$ and $D_e(j,j)$ have the *opposite* sign, then $C_{ij}^{(1)} = 0$ from which we conclude that $C^{(1)}$ has the form

$$C^{(1)} = Q_o C Q_e = \begin{bmatrix} a_0 & b_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_2 & 0 & b_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_3 & 0 & b_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_4 & 0 & b_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_5 & 0 & b_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_7 & b_8 \end{bmatrix}.$$

Analogously, if $D_o(i,i)$ and $D_e(j,j)$ have the *same* sign, then $S_{ij}^{(1)} = 0$ from which we conclude that $S^{(1)}$ has the form

$$S^{(1)} = Q_o S Q_e = \begin{bmatrix} 0 & 0 & f_1 & 0 & 0 & 0 & 0 & 0 \\ e_2 & d_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_3 & 0 & f_3 & 0 & 0 & 0 \\ 0 & e_4 & 0 & d_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_5 & 0 & f_5 & 0 \\ 0 & 0 & 0 & e_6 & 0 & d_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_7 & f_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & e_8 & 0 \end{bmatrix}.$$

Row/column permutations of these matrices result in bidiagonal forms:

$$\begin{aligned} B_C &= C^{(1)}([\,1\,3\,5\,7\,2\,4\,6\,8\,],[\,1\,2\,4\,6\,3\,5\,7\,8\,]) \\ &= \left[\begin{array}{cccc|cccc} a_0 & b_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_2 & b_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_4 & b_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_6 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & b_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_3 & b_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_5 & b_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_7 & b_8 \end{array}\right] \end{aligned}$$

$$B_S = S^{(1)}([\,2\,4\,6\,8\,1\,3\,5\,7\,],[\,1\,2\,4\,6\,3\,5\,7\,8\,])$$

$$= \left[\begin{array}{cccc|cccc} e_2 & d_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e_4 & d_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e_6 & d_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e_8 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & f_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_3 & f_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & d_5 & f_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_7 & f_7 \end{array}\right].$$

It is not hard to verify that a's, b's, d's, e's, and f's are all nonzero and this implies that the singular values of $B_C(1{:}m, 1{:}m)$ and $B_S(1{:}m, 1{:}m)$ are distinct. Since

$$\begin{aligned} \sigma(C) = \sigma(B_C) &= \{\cos(\theta_1/2),\ \cos(\theta_1/2), \ldots, \cos(\theta_m/2),\ \cos(\theta_m/2)\,\} \\ \sigma(S) = \sigma(B_S) &= \{\sin(\theta_1/2),\ \sin(\theta_1/2), \ldots, \sin(\theta_m/2),\ \sin(\theta_m/2)\,\} \end{aligned}$$

we have verified (12.6.5) and (12.6.6).

Problems

P12.6.1 Let $A \in \mathbb{R}^{m\times n}$ and consider the problem of finding the stationary values of

$$R(x,y) = \frac{y^TAx}{\|\,y\,\|_2\|\,x\,\|_2} \qquad y \in \mathbb{R}^m, x \in \mathbb{R}^n$$

subject to the constraints

$$\begin{array}{lll} C^Tx = 0 & C \in \mathbb{R}^{n\times p} & n \ge p \\ D^Ty = 0 & D \in \mathbb{R}^{m\times q} & m \ge q \end{array}$$

Show how to solve this problem by first computing complete orthogonal decompositions of C and D and then computing the SVD of a certain submatrix of a transformed A.

P12.6.2 Suppose $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$. Assume that $\text{rank}(A) = n$ and $\text{rank}(B) = p$. Using the methods of this section, show how to solve

$$\min_{Bx=0} \frac{\|\,b - Ax\,\|_2^2}{\|\,x\,\|_2^2\ + 1} = \min_{Bx=0} \frac{\left\|[\,A,\,b\,]\begin{bmatrix} x \\ -1 \end{bmatrix}\right\|_2^2}{\left\|\begin{bmatrix} x \\ -1 \end{bmatrix}\right\|_2^2}$$

Show that this is a constrained TLS problem. Is there always a solution?

P12.6.3 Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric and that $B \in \mathbb{R}^{p\times n}$ has rank p. Let $d \in \mathbb{R}^p$. Show how to solve the problem of minimizing x^TAx subject to the constraints $\|\,x\,\|_2 = 1$ and $Bx = d$. Indicate when a solution fails to exist.

P12.6.4 Assume that $A \in \mathbb{R}^{n\times n}$ is symmetric, large, and sparse and that $C \in \mathbb{R}^{n\times p}$ is also large and sparse. How can the Lanczos process be used to find the stationary values of

$$r(x) = \frac{x^TAx}{x^Tx}$$

subject to the constraint $C^Tx = 0$? Assume that a sparse QR factorization $C = QR$ is available.

P12.6.5 Relate the eigenvalues and eigenvectors of

$$A = \begin{bmatrix} 0 & A_1 & 0 & 0 \\ 0 & 0 & A_2 & 0 \\ 0 & 0 & 0 & A_3 \\ A_4 & 0 & 0 & 0 \end{bmatrix}.$$

to the eigenvalues and eigenvectors of $\tilde{A} = A_1A_2A_3A_4$. Assume that the diagonal blocks in A are square.

P12.6.6 Prove that if (12.6.2) holds, then (12.6.3) converges to $\lambda_{min}(T)$ montonically from the right.

P12.6.7 Recall from §4.7 that it is possible to compute the inverse of a symmetric positive definite Toeplitz matrix in $O(n^2)$ flops. Use this fact to obtain an initial bracketing interval for (12.6.5) that is based on $\| T^{-1} \|_\infty$ and $\| G^{-1} \|_\infty$.

P12.6.8 A matrix $A \in \mathbb{R}^{n\times n}$ is *centrosymmetric* if it is symmetric and persymmetric, i.e., $A = E_nAE_n$ where $E_n = I_n(:,n:-1:1)$. Show that if $n = 2m$ and Q is the orthogonal matrix

$$Q = \frac{1}{\sqrt{2}}\begin{bmatrix} I_m & I_m \\ E_m & -E_m \end{bmatrix},$$

then

$$Q^TAQ = \begin{bmatrix} A_{11} + A_{12}E_m & 0 \\ 0 & A_{11} - A_{12}E_m \end{bmatrix}$$

where $A_{11} = A(1{:}m, 1{:}m)$ and $A_{12} = A(1{:}m, m+1{:}n)$. Show that if $n = 2m$, then the Schur decomposition of a centrosymmetric matrix can be computed with one-fourth the flops that it takes to compute the Schur decomposition of a symmetric matrix, assuming that the QR algorithm is used in both cases. Repeat the problem if $n = 2m + 1$.

P12.6.9 Suppose $F, G \in \mathbb{R}^{n\times n}$ are symmetric and that

$$Q = \begin{bmatrix} \underset{p}{Q_1} & \underset{n-p}{Q_2} \end{bmatrix}$$

is an n-by-n orthogonal matrix. Show how to compute Q and p so that

$$f(Q,p) = \text{tr}(Q_1^TFQ_1) + \text{tr}(Q_2^TGQ_2)$$

is maximized. Hint: $\text{tr}(Q_1^TFQ_1) + \text{tr}(Q_2^TGQ_2) = \text{tr}(Q_1^T(F-G)Q_1) + \text{tr}(G)$.

P12.6.10 Suppose $A \in \mathbb{R}^{n\times n}$ is given and consider the problem of minimizing $\| A - S \|_F$ over all symmetric positive semidefinite matrices S that have rank r or less. Show that

$$S = \sum_{i=1}^{\min\{k,r\}} \lambda_i q_i q_i^T$$

solves this problem where

$$\frac{A + A^T}{2} = Q\text{diag}(\lambda_1, \ldots, \lambda_n)Q^T$$

is the Schur decomposition of A's symmetric part, $Q = [\, q_1, \ldots, q_n \,]$, and

$$\lambda_1 \geq \cdots \geq \lambda_k > 0 \geq \lambda_{k+1} \geq \cdots \geq \lambda_n.$$

P12.6.11 Verify for general n (even) that H is similar to H_oH_e where these matrices are defined in §12.6.4.

P12.6.12 Verify that the bidiagonal matrices $B_C(1{:}m, 1{:}m)$ and $B_S(1{:}m, 1{:}m)$ in §12.6.4

have nonzero entries on their diagonal and superdiagonal and specify their value.

P12.6.13 A real $2n$-by-$2n$ matrix of the form

$$M = \begin{bmatrix} A & G \\ F & -A^T \end{bmatrix}$$

is *Hamiltonian* if $A \in \mathbb{R}^{n\times n}$ and $F, G \in \mathbb{R}^{n\times n}$ are symmetric. Equivalently, if the orthogonal matrix J is defined by

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix},$$

then $M \in \mathbb{R}^{2n\times 2n}$ is Hamiltonian if and only if $J^TMJ = -M^T$. (a) Show that the eigenvalues of a Hamiltonian matrix come in plus-minus pairs. (b) A matrix $S \in \mathbb{R}^{2n\times 2n}$ is *symplectic* if $J^TSJ = -S^{-T}$. Show that if S is symplectic and M is Hamiltonian, then $S^{-1}MS$ is also Hamiltonian. (c) Show that if $Q \in \mathbb{R}^{2n\times 2n}$ is orthogonal *and* symplectic, then

$$Q = \begin{bmatrix} Q_1 & Q_2 \\ -Q_2 & Q_1 \end{bmatrix}$$

where $Q_1^TQ_1 + Q_2^TQ_2 = I_n$ and $Q_2^TQ_1$ is symmetric. Thus, a Givens rotation of the form $G(i, i+n, \theta)$ is orthogonal symplectic as is the direct sum of n-by-n Householders. (d) Show how to compute a symplectic orthogonal U such that

$$U^TMU = \begin{bmatrix} H & R \\ D & -H^T \end{bmatrix}$$

where H is upper Hessenberg and D is diagonal.

Notes and References for Sec. 12.6

The inverse eigenvalue problems discussed in this §12.6.1 and §12.6.2 appear in the following survey articles:

G.H. Golub (1973). "Some Modified Matrix Eigenvalue Problems," *SIAM Review 15*, 318–44.

D. Boley and G.H. Golub (1987). "A Survey of Matrix Inverse Eigenvalue Problems," *Inverse Problems 3*, 595–622.

References for the stationary value problem include

G.E. Forsythe and G.H. Golub (1965). "On the Stationary Values of a Second-Degree Polynomial on the Unit Sphere," *SIAM J. App. Math. 13*, 1050–68.

G.H. Golub and R. Underwood (1970). "Stationary Values of the Ratio of Quadratic Forms Subject to Linear Constraints," *Z. Angew. Math. Phys. 21*, 318–26.

S. Leon (1994). "Maximizing Bilinear Forms Subject to Linear Constraints," *Lin. Alg. and Its Applic. 210*, 49–58.

An algorithm for minimizing x^TAx where x satisfies $Bx = d$ *and* $\| x \|_2=1$ is presented in

W. Gander, G.H. Golub, and U. von Matt (1991). "A Constrained Eigenvalue Problem," in *Numerical Linear Algebra, Digital Signal Processing, and Parallel Algorithms*, G.H. Golub and P. Van Dooren (eds), Springer-Verlag, Berlin.

Selected papers that discuss a range of inverse eigenvalue problems include

G.H. Golub and J.H. Welsch (1969). "Calculation of Gauss Quadrature Rules," *Math. Comp. 23*, 221–30.

S. Friedland (1975). "On Inverse Multiplicative Eigenvalue Problems for Matrices," *Lin. Alg. and Its Applic. 12*, 127–38.

D.L. Boley and G.H. Golub (1978). "The Matrix Inverse Eigenvalue Problem for Periodic Jacobi Matrices," in *Proc. Fourth Symposium on Basic Problems of Numerical Mathematics*, Prague, pp. 63–76.
W.E. Ferguson (1980). "The Construction of Jacobi and Periodic Jacobi Matrices with Prescribed Spectra," *Math. Comp. 35*, 1203–1220.
J. Kautsky and G.H. Golub (1983). "On the Calculation of Jacobi Matrices," *Lin. Alg. and Its Applic. 52/53*, 439–456.
D. Boley and G.H. Golub (1984). "A Modified Method for Restructuring Periodic Jacobi Matrices," *Math. Comp. 42*, 143–150.
W.B. Gragg and W.J. Harrod (1984). "The Numerically Stable Reconstruction of Jacobi Matrices from Spectral Data," *Numer. Math. 44*, 317–336.
S. Friedland, J. Nocedal, and M.L. Overton (1987). "The Formulation and Analysis of Numerical Methods for Inverse Eigenvalue Problems," *SIAM J. Numer. Anal. 24*, 634–667.
M.T. Chu (1992). "Numerical Methods for Inverse Singular Value Problems," *SIAM J. Num. Anal. 29*, 885–903.
G. Ammar and G. He (1995). "On an Inverse Eigenvalue Problem for Unitary Matrices," *Lin. Alg. and Its Applic. 218*, 263–271.
H. Zha and Z. Zhang (1995). "A Note on Constructing a Symmetric Matrix with Specified Diagonal Entries and Eigenvalues," *BIT 35*, 448–451.

Various Toeplitz eigenvalue computations are presented in

G. Cybenko and C. Van Loan (1986). "Computing the Minimum Eigenvalue of a Symmetric Positive Definite Toeplitz Matrix," *SIAM J. Sci. and Stat. Comp. 7*, 123–131.
W.F. Trench (1989). "Numerical Solution of the Eigenvalue Problem for Hermitian Toeplitz Matrices," *SIAM J. Matrix Anal. Appl. 10*, 135–146.
L. Reichel and L.N. Trefethen (1992). "Eigenvalues and Pseudo-eigenvalues of Toeplitz Matrices," *Lin. Alg. and Its Applic. 162/163/164*, 153–186.
S.L. Handy and J.L. Barlow (1994). "Numerical Solution of the Eigenproblem for Banded, Symmetric Toeplitz Matrices," *SIAM J. Matrix Anal. Appl. 15*, 205–214.

Unitary/orthogonal eigenvalue problems are treated in

H. Rutishauser (1966). "Bestimmung der Eigenwerte Orthogonaler Matrizen," *Numer. Math. 9*, 104–108.
P.J. Eberlein and C.P. Huang (1975). "Global Convergence of the QR Algorithm for Unitary Matrices with Some Results for Normal Matrices," *SIAM J. Numer. Anal. 12*, 421–453.
G. Cybenko (1985). "Computing Pisarenko Frequency Estimates," in *Proceedings of the Princeton Conference on Information Science and Systems*, Dept. of Electrical Engineering, Princeton University.
W. B. Gragg (1986). "The QR Algorithm for Unitary Hessenberg Matrices," *J. Comp. Appl. Math. 16*, 1–8.
G.S. Ammar, W.B. Gragg, and L. Reichel (1985). "On the Eigenproblem for Orthogonal Matrices," *Proc. IEEE Conference on Decision and Control*, 1963 1966.
W.B. Gragg and L. Reichel (1990). "A Divide and Conquer Method for Unitary and Orthogonal Eigenproblems," *Numer. Math. 57*, 695–718.

Hamiltonian eigenproblems (see P12.6.13) occur throughout optimal control theory and are very important.

C.C. Paige and C. Van Loan (1981). "A Schur Decomposition for Hamiltonian Matrices," *Lin. Alg. and Its Applic. 41*, 11–32.
C. Van Loan (1984). "A Symplectic Method for Approximating All the Eigenvalues of a Hamiltonian Matrix," *Lin. Alg. and Its Applic. 61*, 233–252.

R. Byers (1986) "A Hamiltonian QR Algorithm," *SIAM J. Sci. and Stat. Comp. 7*, 212–229.
V. Mehrmann (1988). "A Symplectic Orthogonal Method for Single Input or Single Output Discrete Time Optimal Quadratic Control Problems," *SIAM J. Matrix Anal. Appl. 9*, 221–247.
G. Ammar and V. Mehrmann (1991). "On Hamiltonian and Symplectic Hessenberg Forms," *Lin.Alg. and Its Application 149*, 55–72.
A. Bunse-Gerstner, R. Byers, and V. Mehrmann (1992). "A Chart of Numerical Methods for Structured Eigenvalue Problems," *SIAM J. Matrix Anal. Appl. 13*, 419–453.

Other papers on modified/structured eigenvalue problems include

A. Bunse-Gerstner and W.B. Gragg (1988). "Singular Value Decompositions of Complex Symmetric Matrices," *J. Comp. Applic. Math. 21*, 41–54.
R. Byers (1988). "A Bisection Method for Measuring the Distance of a Stable Matrix to the Unstable Matrices," *SIAM J. Sci. Stat. Comp. 9*, 875–881.
J.W. Demmel and W. Gragg (1993). "On Computing Accurate Singular Values and Eigenvalues of Matrices with Acyclic Graphs," *Lin. Alg. and Its Applic. 185*, 203–217.
A. Bunse-Gerstner, R. Byers, and V. Mehrmann (1993). "Numerical Methods for Simultaneous Diagonalization," *SIAM J. Matrix Anal. Appl. 14*, 927–949.

Bibliography

J.O. Aasen (1971). "On the Reduction of a Symmetric Matrix to Tridiagonal Form," *BIT 11*, 233–42.

N.N. Abdelmalek (1971). "Roundoff Error Analysis for Gram-Schmidt Method and Solution of Linear Least Squares Problems," *BIT 11*, 345–68.

G.E. Adams, A.W. Bojanczyk, and F.T. Luk (1994). "Computing the PSVD of Two 2x2 Triangular Matrices," *SIAM J. Matrix Anal. Appl. 15*, 366–382.

L. Adams (1985). "m-step Preconditioned Congugate Gradient Methods," *SIAM J. Sci. and Stat. Comp. 6*, 452–463.

L. Adams and P. Arbenz (1994). "Towards a Divide and Conquer Algorithm for the Real Nonsymmetric Eigenvalue Problem," *SIAM J. Matrix Anal. Appl. 15*, 1333–1353.

L. Adams and T. Crockett (1984). "Modeling Algorithm Execution Time on Processor Arrays," *Computer 17*, 38–43.

L. Adams and H. Jordan (1986). "Is SOR Color-Blind?" *SIAM J. Sci. Stat. Comp. 7*, 490–506.

S.T. Alexander, C.T. Pan, and R.J. Plemmons (1988). "Analysis of a Recursive Least Squares Hyperbolic Rotation Algorithm for Signal Processing," *Lin. Alg. and Its Applic. 98*, 3–40.

E.L. Allgower (1973). "Exact Inverses of Certain Band Matrices," *Numer. Math. 21*, 279–84.

A.R. Amir-Moez (1965). *Extremal Properties of Linear Transformations and Geometry of Unitary Spaces*, Texas Tech University Mathematics Series, no. 243, Lubbock, Texas.

G.S. Ammar and W.B. Gragg (1988). "Superfast Solution of Real Positive Definite Toeplitz Systems," *SIAM J. Matrix Anal. Appl. 9*, 61–76.

G.S. Ammar, W.B. Gragg, and L. Reichel (1985). "On the Eigenproblem for Orthogonal Matrices," *Proc. IEEE Conference on Decision and Control*, 1963–1966.

G.S. Ammar and G. He (1995). "On an Inverse Eigenvalue Problem for Unitary Matrices," *Lin. Alg. and Its Applic. 218*, 263–271.

G.S. Ammar and V. Mehrmann (1991). "On Hamiltonian and Symplectic Hessenberg Forms," *Lin.Alg. and Its Applic. 149*, 55–72.

P. Amodio and L. Brugnano (1995). "The Parallel QR Factorization Algorithm for Tridiagonal Linear Systems," *Parallel Computing 21*, 1097–1110.

C. Ancourt, F. Coelho, F. Irigoin, and R. Keryell (1993). "A Linear Algebra Framework for Static HPF Code Distribution," *Proceedings of the 4th Workshop on Compilers for Parallel Computers*, Delft, The Netherlands.

A.A. Anda and H. Park (1994). "Fast Plane Rotations with Dynamic Scaling," *SIAM J. Matrix Anal. Appl. 15*, 162–174.

E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen (1995). *LAPACK Users' Guide, Release 2.0, 2nd ed.*, SIAM Publications, Philadelphia, PA.

E. Anderson, Z. Bai, and J. Dongarra (1992). "Generalized QR Factorization and Its Application," *Lin. Alg. and Its Applic. 162/163/164*, 243–271.

N. Anderson and I. Karasalo (1975). "On Computing Bounds for the Least Singular Value of a Triangular Matrix," *BIT 15*, 1–4.

P. Anderson and G. Loizou (1973). "On the Quadratic Convergence of an Algorithm Which Diagonalizes a Complex Symmetric Matrix," *J. Inst. Math. Applic.* *12*, 261–71.

P. Anderson and G. Loizou (1976). "A Jacobi-Type Method for Complex Symmetric Matrices (Handbook)," *Numer. Math.* *25*, 347–63.

H.C. Andrews and J. Kane (1970). "Kronecker Matrices, Computer Implementation, and Generalized Spectra," *J. Assoc. Comput. Mach.* *17*, 260–268.

P. Arbenz, W. Gander, and G.H. Golub (1988). "Restricted Rank Modification of the Symmetric Eigenvalue Problem: Theoretical Considerations," *Lin. Alg. and Its Applic.* *104*, 75–95.

P. Arbenz and G.H. Golub (1988). "On the Spectral Decomposition of Hermitian Matrices Subject to Indefinite Low Rank Perturbations with Applications," *SIAM J. Matrix Anal. Appl.* *9*, 40–58.

T.A. Arias, A. Edelman, and S. Smith (1996). "Conjugate Gradient and Newton's Method on the Grassman and Stiefel Manifolds," to appear in *SIAM J. Matrix Anal. Appl.*

M. Arioli, J. Demmel, and I. Duff (1989). "Solving Sparse Linear Systems with Sparse Backward Error," *SIAM J. Matrix Anal. Appl.* *10*, 165–190.

M. Arioli, I.S. Duff, and D. Ruiz (1992). "Stopping Criteria for Iterative Solvers," *SIAM J. Matrix Anal. Appl.* *13*, 138–144.

M. Arioli and A. Laratta (1985). "Error Analysis of an Algorithm for Solving an Underdetermined System," *Numer. Math.* *46*, 255–268.

M. Arioli and F. Romani (1985). "Relations Between Condition Numbers and the Convergence of the Jacobi Method for Real Positive Definite Matrices," *Numer. Math.* *46*, 31–42.

W.F. Arnold and A.J. Laub (1984). "Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations," *Proc. IEEE* *72*, 1746–1754.

W.E. Arnoldi (1951). "The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem," *Quarterly of Applied Mathematics* *9*, 17 29.

S. Ashby (1987). "Polynomial Preconditioning for Conjugate Gradient Methods," Ph.D. Thesis, Dept. of Computer Science, University of Illinois.

S. Ashby, T. Manteuffel, and J. Otto (1992). "A Comparison of Adaptive Chebyshev and Least Squares Polynomial Preconditioning for Hermitian Positive Definite Linear Systems," *SIAM J. Sci. Stat. Comp.* *13*, 1–29.

S. Ashby, T. Manteuffel, and P. Saylor (1989). "Adaptive Polynomial Preconditioning for Hermitian Indefinite Linear Systems," *BIT* *29*, 583–609.

S. Ashby, T. Manteuffel, and P. Saylor (1990). "A Taxonomy of Conjugate Gradient Methods," *SIAM J. Num. Anal.* *27*, 1542–1568.

C.C. Ashcraft and R. Grimes (1988). "On Vectorizing Incomplete Factorization and SSOR Preconditioners," *SIAM J. Sci. and Stat. Comp.* *9*, 122 151.

G. Auchmuty (1991). "A Posteriori Error Estimates for Linear Equations," *Numer. Math.* *61*, 1–6.

O. Axelsson (1977). "Solution of Linear Systems of Equations: Iterative Methods," in *Sparse Matrix Techniques: Copenhagen*, 1976, ed. V.A. Barker, Springer-Verlag, Berlin.

O. Axelsson (1980). "Conjugate Gradient Type Methods for Unsymmetric and Inconsistent Systems of Linear Equations," *Lin. Alg. and Its Applic.* *29*, 1–16.

O. Axelsson (1985). "Incomplete Block Matrix Factorization Preconditioning Methods. The Ultimate Answer?", *J. Comput. Appl. Math.* *12&13*, 3–18.

O. Axelsson (1985). "A Survey of Preconditioned Iterative Methods for Linear Systems of Equations," *BIT* *25*, 166–187.

O. Axelsson (1986). "A General Incomplete Block Matrix Factorization Method," *Lin. Alg. Appl.* *74*, 179–190.

O. Axelsson, ed. (1989). "Preconditioned Conjugate Gradient Methods," *BIT 29:4*.

O. Axelsson (1994). *Iterative Solution Methods*, Cambridge University Press.

O. Axelsson and V. Eijkhout (1989). "Vectorizable Preconditioners for Elliptic Difference Equations in Three Space Dimensions," *J. Comput. Appl. Math.* *27*, 299–321.

O. Axelsson and G. Lindskog (1986). "On the Rate of Convergence of the Preconditioned Conjugate Gradient Method," *Numer. Math.* *48*, 499–523.

O. Axelsson and B. Polman (1986). "On Approximate Factorization Methods for Block Matrices Suitable for Vector and Parallel Processors," *Lin. Alg. and Its Applic.* *77*, 3–26.

O. Axelsson and P. Vassilevski (1989). "Algebraic Multilevel Preconditioning Methods I," *Numer. Math.* *56*, 157–177.

O. Axelsson and P. Vassilevski (1990). "Algebraic Multilevel Preconditioning Methods II," *SIAM J. Numer. Anal.* *27*. 1569–1590.

Z. Bai (1988). "Note on the Quadratic Convergence of Kogbetliantz's Algorithm for Computing the Singular Value Decomposition," *Lin. Alg. and Its Applic.* *104*, 131–140.

Z. Bai (1994). "Error Analysis of the Lanczos Algorithm for Nonsymmetric Eigenvalue Problem," *Math. Comp.* *62*, 209–226.

Z. Bai and J.W. Demmel (1989). "On a Block Implementation of Hessenberg Multishift QR Iteration," *Int'l J. of High Speed Comput.* *1*, 97–112.

Z. Bai and J.W. Demmel (1993). "On Swapping Diagonal Blocks in Real Schur Form," *Lin. Alg. and Its Applic.* *186*, 73–95.

Z. Bai, J.W. Demmel, and A. McKenney (1993). "On Computing Condition Numbers for the Nonsymmetric Eigenproblem," *ACM Trans. Math. Soft.* *19*, 202–223.

Z. Bai, D. Hu, and L. Reichel (1994). "A Newton Basis GMRES Implementation," *IMA J. Num. Anal.* *14*, 563–581.

Z. Bai and H. Zha (1993). "A New Preprocessing Algorithm for the Computation of the Generalized Singular Value Decomposition," *SIAM J. Sci. Comp.* *14*, 1007–1012.

D.H. Bailey (1988). "Extra High Speed Matrix Multiplication on the Cray-2," *SIAM J. Sci. and Stat. Comp.* *9*, 603–607.

D.H. Bailey (1993). "Algorithm 719: Multiprecision Translation and Execution of FORTRAN Programs," *ACM Trans. Math. Soft.* *19*, 288–319.

D.H. Bailey, K.Lee, and H.D. Simon (1991). "Using Strassen's Algorithm to Accelerate the Solution of Linear Systems," *J. Supercomputing* *4*, 357–371.

D.H. Bailey, H.D. Simon, J. T. Barton, M.J. Fouts (1989). "Floating Point Arithmetic in Future Supercomputers," *Int'l J. Supercomputing Appl.* *3*, 86–90.

I.Y. Bar-Itzhack (1975). "Iterative Optimal Orthogonalization of the Strapdown Matrix," *IEEE Trans. Aerospace and Electronic Systems* *11*, 30–37.

J.L. Barlow (1986). "On the Smallest Positive Singular Value of an M-Matrix with Applications to Ergodic Markov Chains," *SIAM J. Alg. and Disc. Struct.* *7*, 414–424.

J.L. Barlow (1988). "Error Analysis and Implementation Aspects of Deferred Correction for Equality Constrained Least-Squares Problems," *SIAM J. Num. Anal.* *25*, 1340–1358.

J.L. Barlow (1993). "Error Analysis of Update Methods for the Symmetric Eigenvalue Problem," *SIAM J. Matrix Anal. Appl.* *14*, 598–618.

J.L. Barlow and J. Demmel (1990). "Computing Accurate Eigensystems of Scaled Diagonally Dominant Matrices," *SIAM J. Numer. Anal.* *27*, 762–791.

J.L. Barlow and S.L. Handy (1988). "The Direct Solution of Weighted and Equality Constrained Least-Squares Problems," *SIAM J. Sci. Stat. Comp.* *9*, 704–716.

J.L. Barlow, M.M. Monahemi, and D.P. O'Leary (1992). "Constrained Matrix Sylvester Equations," *SIAM J. Matrix Anal. Appl.* *13*, 1–9.

J.L. Barlow, N.K. Nichols, and R.J. Plemmons (1988). "Iterative Methods for Equality Constrained Least Squares Problems," SIAM J. Sci. and Stat. Comp. 9, 892–906.

J.L. Barlow and U.B. Vemulapati (1992). "Rank Detection Methods for Sparse Matrices," *SIAM J. Matrix. Anal. Appl.* *13*, 1279–1297.

J.L. Barlow and U.B. Vemulapati (1992). "A Note on Deferred Correction for Equality Constrained Least Squares Problems," *SIAM J. Num. Anal.* *29*, 249–256.

S. Barnett and C. Storey (1968). "Some Applications of the Lyapunov Matrix Equation," *J. Inst. Math. Applic. 4*, 33–42.

R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. van der Vorst (1993). *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM Publications, Philadelphia, PA.

A. Barrlund (1991). "Perturbation Bounds for the LDL^T and LU Decompositions," *BIT 31*, 358–363.

A. Barrlund (1994). "Perturbation Bounds for the Generalized QR Factorization," *Lin. Alg. and Its Applic. 207*, 251–271.

R.H. Bartels (1971). "A Stabilization of the Simplex Method," *Numer. Math. 16*, 414-434.

R.H. Bartels, A.R. Conn, and C. Charalambous (1978). "On Cline's Direct Method for Solving Overdetermined Linear Systems in the L_∞ Sense," *SIAM J. Num. Anal. 15*, 255–70.

R.H. Bartels and G.W. Stewart (1972). "Solution of the Equation $AX + XB = C$," *Comm. ACM 15*, 820–26.

S.G. Bartels and D.J. Higham (1992). "The Structured Sensitivity of Vandermonde-Like Systems," *Numer. Math. 62*, 17–34.

W. Barth, R.S. Martin, and J.H. Wilkinson (1967). "Calculation of the Eigenvalues of a Symmetric Tridiagonal Matrix by the Method of Bisection," *Numer. Math. 9*, 386–93. See also Wilkinson and Reinsch (1971, 249–256).

V. Barwell and J.A. George (1976). "A Comparison of Algorithms for Solving Symmetric Indefinite Systems of Linear Equations," *ACM Trans. Math. Soft. 2*, 242–51.

K.J. Bathe and E.L. Wilson (1973). "Solution Methods for Eigenvalue Problems in Structural Mechanics," *Int. J. Numer. Meth. Eng. 6*, 213-26.

S. Batterson (1994). "Convergence of the Francis Shifted QR Algorithm on Normal Matrices," *Lin. Alg. and Its Applic. 207*, 181–195.

S. Batterson and J. Smillie (1989). "The Dynamics of Rayleigh Quotient Iteration," *SIAM J. Num. Anal. 26*, 624–636.

D. Bau, I. Kodukula, V. Kotlyar, K. Pingali, and P. Stodghill (1993). "Solving Alignment Using Elementary Linear Algebra," in *Proceedings of the 7th International Workshop on Languages and Compilers for Parallel Computing*, Lecture Notes in Computer Science 892, Springer-Verlag, New York, 46–60.

F.L. Bauer (1963). "Optimally Scaled Matrices," *Numer. Math. 5*, 73–87.

F.L. Bauer (1965). "Elimination with Weighted Row Combinations for Solving Linear Equations and Least Squares Problems," *Numer. Math. 7*, 338–52. See also Wilkinson and Reinsch (1971, 119–33).

F.L. Bauer and C.T. Fike (1960). "Norms and Exclusion Theorems," *Numer. Math. 2*, 123–44.

F.L. Bauer and C. Reinsch (1968). "Rational QR Transformations with Newton Shift for Symmetric Tridiagonal Matrices," *Numer. Math. 11*, 264–72. See also Wilkinson and Reinsch (1971, pp.257–65).

F.L. Bauer and C. Reinsch (1971). "Inversion of Positive Definite Matrices by the Gauss-Jordan Method," in *Handbook for Automatic Computation Vol. 2, Linear Algebra*, J.H. Wilkinson and C. Reinsch, eds. Springer-Verlag, New York, 45–49.

C. Bavely and G.W. Stewart (1979). "An Algorithm for Computing Reducing Subspaces by Block Diagonalization," *SIAM J. Num. Anal. 16*, 359–67.

C. Beattie and D.W. Fox (1989). "Localization Criteria and Containment for Rayleigh Quotient Iteration," *SIAM J. Matrix Anal. Appl. 10*, 80–93.

R. Beauwens and P. de Groen, eds. (1992). *Iterative Methods in Linear Algebra*, Elsevier (North-Holland), Amsterdam.

T Beelen and P. Van Dooren (1988). "An Improved Algorithm for the Computation of Kronecker's Canonical Form of a Singular Pencil," *Lin. Alg. and Its Applic. 105*, 9–65.

R. Bellman (1970). *Introduction to Matrix Analysis*, 2nd ed., McGraw-Hill, New York.

E. Beltrami (1873). "Sulle Funzioni Bilineari," *Gionale di Mathematiche 11*, 98–106.

A. Berman and A. Ben-Israel (1971). "A Note on Pencils of Hermitian or Symmetric Matrices," *SIAM J. Applic. Math. 21*, 51-54.

A. Berman and R.J. Plemmons (1979). *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York.

M.J.M. Bernal and J.H. Verner (1968). "On Generalizing of the Theory of Consistent Orderings for Successive Over-Relaxation Methods," *Numer. Math. 12*, 215–22.

J. Berntsen (1989). "Communication Efficient Matrix Multiplication on Hypercubes," *Parallel Computing 12*, 335–342.

M.W. Berry, J.J. Dongarra, and Y. Kim (1995). "A Parallel Algorithm for the Reduction of a Nonsymmetric Matrix to Block Upper Hessenberg Form," *Parallel Computing 21*, 1189–1211.

M.W. Berry and G.H. Golub (1991). "Estimating the Largest Singular Values of Large Sparse Matrices via Modified Moments," *Numerical Algorithms 1*, 353-374.

M.W. Berry and A. Sameh (1986). "Multiprocessor Jacobi Algorithms for Dense Symmetric Eigenvalue and Singular Value Decompositions," in *Proc. International Conference on Parallel Processing*, 433–440.

D.P. Bertsekas and J. N. Tsitsiklis (1989). *Parallel and Distributed Computation: Numerical Methods*, Prentice Hall, Englewood Cliffs, NJ.

R. Bevilacqua, B. Codenotti, and F. Romani (1988). "Parallel Solution of Block Tridiagonal Linear Systems," *Lin.Alg. and Its Applic. 104*, 39–57.

C.H. Bischof (1987). "The Two-Sided Block Jacobi Method on Hypercube Architectures," in *Hypercube Multiprocessors*, ed. M.T. Heath, SIAM Publications, Philadelphia, PA.

C.H. Bischof (1988). "QR Factorization Algorithms for Coarse Grain Distributed Systems," PhD Thesis, Dept. of Computer Science, Cornell University, Ithaca, NY.

C.H. Bischof (1989). "Computing the Singular Value Decomposition on a Distributed System of Vector Processors," *Parallel Computing 11*, 171–186.

C.H. Bischof (1990). "Incremental Condition Estimation," *SIAM J. Matrix Anal. Appl. 11*, 644–659.

C.H. Bischof (1990). "Incremental Condition Estimation for Sparse Matrices," *SIAM J. Matrix Anal. Appl. 11*, 312-322.

C.H. Bischof and P.C. Hansen (1992). "A Block Algorithm for Computing Rank-Revealing QR Factorizations," *Numerical Algorithms 2*, 371-392.

C.H. Bischof and G.M. Shroff (1992). "On Updating Signal Subspaces," *IEEE Trans. Signal Proc. 40*, 96–105.

C.H. Bischof and C. Van Loan (1986). "Computing the SVD on a Ring of Array Processors," in *Large Scale Eigenvalue Problems*, eds. J. Cullum and R. Willoughby, North Holland, 51-66.

C.H. Bischof and C. Van Loan (1987). "The WY Representation for Products of Householder Matrices," *SIAM J. Sci. and Stat. Comp. 8*, s2–s13.

Å. Björck (1967). "Iterative Refinement of Linear Least Squares Solutions I," *BIT 7*, 257–78.

A. Björck (1967). "Solving Linear Least Squares Problems by Gram-Schmidt Orthogonalization," *BIT 7*, 1-21.

Å. Björck (1968). "Iterative Refinement of Linear Least Squares Solutions II," *BIT 8*, 8–30.

Å. Björck (1984). "A General Updating Algorithm for Constrained Linear Least Squares Problems," *SIAM J. Sci. and Stat. Comp. 5*, 394–402.

Å. Björck (1987). "Stability Analysis of the Method of Seminormal Equations for Linear Least Squares Problems," *Linear Alg. and Its Applic. 88/89*, 31–48.

Å. Björck (1991). "Component-wise Perturbation Analysis and Error Bounds for Linear Least Squares Solutions," *BIT 31*, 238-244.

Å. Björck (1992). "Pivoting and Stability in the Augmented System Method," *Proceedings of the 14th Dundee Conference*, D.F. Griffiths and G.A. Watson (eds), Longman Scientific and Technical, Essex, U.K.

A. Björck (1994). "Numerics of Gram-Schmidt Orthogonalization," *Lin. Alg. and Its Applic. 197/198*, 297–316.

Å. Björck (1996). *Numerical Methods for Least Squares Problems*, SIAM Publications, Philadelphia, PA.

A. Björck and C. Bowie (1971). "An Iterative Algorithm for Computing the Best Estimate of an Orthogonal Matrix," *SIAM J. Num. Anal. 8*, 358–64.

Å. Björck and I.S. Duff (1980). "A Direct Method for the Solution of Sparse Linear Least Squares Problems," *Lin. Alg. and Its Applic. 34*, 43–67.

A. Björck and T. Elfving (1973). "Algorithms for Confluent Vandermonde Systems," *Numer. Math. 21*, 130–37.

Å. Björck and G.H. Golub (1967). "Iterative Refinement of Linear Least Squares Solutions by Householder Transformation," *BIT 7*, 322–37.

A. Björck and G.H. Golub (1973). "Numerical Methods for Computing Angles Between Linear Subspaces," *Math. Comp. 27*, 579–94.

Å. Björck, E. Grimme, and P. Van Dooren (1994). "An Implicit Shift Bidiagonalization Algorithm for Ill-Posed Problems," *BIT 34*, 510–534.

Å. Björck and S. Hammarling (1983). "A Schur Method for the Square Root of a Matrix," *Lin. Alg. and Its Applic. 52/53*, 127–140.

P. Bjørstad, F. Manne, T.Sørevik, and M. Vajteršic (1992). "Efficient Matrix Multiplication on SIMD Computers," *SIAM J. Matrix Anal. Appl. 13*, 386–401.

Å. Björck and C.C. Paige (1992). "Loss and Recapture of Orthogonality in the Modified Gram-Schmidt Algorithm," *SIAM J. Matrix Anal. Appl. 13*, 176–190.

Å. Björck and C.C. Paige (1994). "Solution of Augmented Linear Systems Using Orthogonal Factorizations," *BIT 34*, 1–24.

Å. Björck, H. Park, and L. Eldén (1994). "Accurate Downdating of Least Squares Solutions," *SIAM J. Matrix Anal. Appl. 15*, 549–568.

A. Björck and V. Pereyra (1970). "Solution of Vandermonde Systems of Equations," *Math. Comp. 24*, 893–903.

Å. Björck, R.J. Plemmons, and H. Schneider, eds. (1981). *Large-Scale Matrix Problems*, North-Holland, New York.

J.M. Blue (1978). "A Portable FORTRAN Program to Find the Euclidean Norm of a Vector," *ACM Trans. Math. Soft. 4*, 15–23.

E. Bodewig (1959). *Matrix Calculus*, North Holland, Amsterdam.

Z. Bohte (1975). "Bounds for Rounding Errors in the Gaussian Elimination for Band Systems," *J. Inst. Math. Applic. 16*, 133–42.

A.W. Bojanczyk, R.P. Brent, and F.R. de Hoog (1986). "QR Factorization of Toeplitz Matrices," *Numer. Math. 49*, 81–94.

A.W. Bojanczyk, R.P. Brent, F.R. de Hoog, and D.R. Sweet (1995). "On the Stability of the Bareiss and Related Toeplitz Factorization Algorithms," *SIAM J. Matrix Anal. Appl. 16*, 40–57.

A.W. Bojanczyk, R.P. Brent, P. Van Dooren, and F.R. de Hoog (1987). "A Note on Downdating the Cholesky Factorization," *SIAM J. Sci. and Stat. Comp. 8*, 210–221.

A.W. Bojanczyk and G. Cybenko, eds. (1995). *Linear Algebra for Signal Processing*, IMA Volumes in Mathematics and Its Applications, Springer-Verlag, New York.

A.W. Bojanczyk, R. Onn, and A.O. Steinhardt (1993). "Existence of the Hyperbolic Singular Value Decomposition," *Lin. Alg. and Its Applic. 185*, 21–30.

D.L. Boley and G.H. Golub (1978). "The Matrix Inverse Eigenvalue Problem for Periodic Jacobi Matrices," in *Proc. Fourth Symposium on Basic Problems of Numerical Mathematics*, Prague, pp. 63–76.

D.L. Boley and G.H. Golub (1984). "A Modified Method for Restructuring Periodic Jacobi Matrices," *Math. Comp. 42*, 143–150.

D.L. Boley and G.H. Golub (1984). "The Lanczos-Arnoldi Algorithm and Controllability," *Syst. Control Lett. 4*, 317–324.

D.L. Boley and G.H. Golub (1987). "A Survey of Matrix Inverse Eigenvalue Problems," *Inverse Problems 3*, 595–622.

H. Bolz and W. Niethammer (1988). "On the Evaluation of Matrix Functions Given by Power Series," *SIAM J. Matrix Anal. Appl.* *9*, 202–209.

S. Bondeli and W. Gander (1994). "Cyclic Reduction for Special Tridiagonal Systems," *SIAM J. Matrix Anal. Appl.* *15*, 321–330.

J. Boothroyd and P.J. Eberlein (1968). "Solution to the Eigenproblem by a Norm-Reducing Jacobi-Type Method (Handbook)," *Numer. Math.* *11*, 1–12. See also Wilkinson and Reinsch (1971, pp.327–38).

H.J. Bowdler, R.S. Martin, G. Peters, and J.H. Wilkinson (1966). "Solution of Real and Complex Systems of Linear Equations," *Numer. Math.* *8*, 217–234. (See also Wilkinson and Reinsch (1971, 93–110).

H.J. Bowdler, R.S. Martin, C. Reinsch, and J.H. Wilkinson (1968). "The QR and QL Algorithms for Symmetric Matrices," *Numer. Math.* *11*, 293–306. See also Wilkinson and Reinsch (1971, pp.227–240).

J.H. Bramble, J.E. Pasciak, and A.H. Schatz (1986). "The construction of Preconditioners for Elliptic Problems by Substructuring I," *Math. Comp.* *47*, 103–134.

J.H. Bramble, J.E. Pasciak, and A.H. Schatz (1986). "The construction of Preconditioners for Elliptic Problems by Substructuring II," *Math. Comp.* *49*, 1–17.

R. Bramley and A. Sameh (1992). "Row Projection Methods for Large Nonsymmetric Linear Systems," *SIAM J. Sci. Statist. Comput.* *13*, 168–193.

R.P. Brent (1970). "Error Analysis of Algorithms for Matrix Multiplication and Triangular Decomposition Using Winograd's Identity," *Numer. Math.* *16*, 145–156.

R.P. Brent (1978). "A Fortran Multiple Precision Arithmetic Package," *ACM Trans. Math. Soft.* *4*, 57–70.

R.P. Brent (1978). "Algorithm 524 MP, a Fortran Multiple Precision Arithmetic Package," *ACM Trans. Math. Soft.* *4*, 71–81.

R.P. Brent and F.T. Luk (1982) "Computing the Cholesky Factorization Using a Systolic Architecture," *Proc. 6th Australian Computer Science Conf.* 295–302.

R.P. Brent and F.T. Luk (1985). "The Solution of Singular Value and Symmetric Eigenvalue Problems on Multiprocessor Arrays," *SIAM J. Sci. and Stat. Comp.* *6*, 69–84.

R.P. Brent, F.T. Luk, and C. Van Loan (1985). "Computation of the Singular Value Decomposition Using Mesh Connected Processors," *J. VLSI Computer Systems 1*, 242–270.

C. Brezinski and M. Redivo-Zaglia (1994). "Treatment of Near-Breakdown in the CGS Algorithms," *Numer. Alg.* 7, 33–73.

C. Brezinski and M. Redivo-Zaglia (1995). "Look-Ahead in BiCGSTAB and Other Product-Type Methods for Linear Systems," *BIT 35*, 169–201.

C. Brezinski and H. Sadok (1991). "Avoiding Breakdown in the CGS Algorithm," *Numer. Alg.* *1*, 199–206.

C. Brezinski, M. Zaglia, and H. Sadok (1991). "Avoiding Breakdown and Near Breakdown in Lanczos Tyoe Algorithms," *Numer. Alg.* *1*, 261–284.

C. Brezinski, M. Zaglia, and H. Sadok (1992). "A Breakdown Free Lanczos Type Algorithm for Solving Linear Systems," *Numer. Math.* *63*, 29–38.

K.W. Brodlie and M.J.D. Powell (1975). "On the Convergence of Cyclic Jacobi Methods," *J. Inst. Math. Applic.* *15*, 279–87.

J.D. Brown, M.T. Chu, D.C. Ellison, and R.J. Plemmons, eds. (1994). *Proceedings of the Cornelius Lanczos International Centenary Conference*, SIAM Publications, Philadelphia, PA.

C.G. Broyden (1973). "Some Condition Number Bounds for the Gaussian Elimination Process," *J. Inst. Math. Applic.* *12*, 273–86.

A. Buckley (1974). "A Note on Matrices $A = I + H$, H Skew-Symmetric," *Z. Angew. Math. Mech.* *54*, 125–26.

A. Buckley (1977). "On the Solution of Certain Skew-Symmetric Linear Systems," *SIAM J. Num. Anal.* *14*, 566–70.

J.R. Bunch (1971). "Analysis of the Diagonal Pivoting Method," *SIAM J. Num. Anal.* *8*, 656–680.

J.R. Bunch (1971). "Equilibration of Symmetric Matrices in the Max-Norm," *J. ACM 18*, 566–72.

J.R. Bunch (1974). "Partial Pivoting Strategies for Symmetric Matrices," *SIAM J. Num. Anal. 11*, 521 528.

J.R. Bunch (1976). "Block Methods for Solving Sparse Linear Systems," in *Sparse Matrix Computations*, J.R. Bunch and D.J. Rose (eds), Academic Press, New York.

J.R. Bunch (1982). "A Note on the Stable Decomposition of Skew Symmetric Matrices," *Math. Comp. 158*, 475 480.

J.R. Bunch (1985). "Stability of Methods for Solving Toeplitz Systems of Equations," *SIAM J. Sci. Stat. Comp. 6*, 349-364.

J.R. Bunch, J.W. Demmel, and C.F. Van Loan (1989). "The Strong Stability of Algorithms for Solving Symmetric Linear Systems," *SIAM J. Matrix Anal. Appl. 10*, 494–499.

J.R. Bunch and L. Kaufman (1977). "Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems," *Math. Comp. 31*, 162–79.

J.R. Bunch, L. Kaufman, and B.N. Parlett (1976). "Decomposition of a Symmetric Matrix," *Numer. Math. 27*, 95–109.

J.R. Bunch, C.P. Nielsen, and D.C. Sorensen (1978). "Rank-One Modification of the Symmetric Eigenproblem," *Numer. Math. 31*, 31–48.

J.R. Bunch and B.N. Parlett (1971). "Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations," *SIAM J. Num. Anal. 8*, 639–55.

J.R. Bunch and D.J. Rose, eds. (1976). *Sparse Matrix Computations*, Academic Press, New York.

O. Buneman (1969). "A Compact Non-Iterative Poisson Solver," Report 294, Stanford University Institute for Plasma Research, Stanford, California.

A. Bunse-Gerstner (1984). "An Algorithm for the Symmetric Generalized Eigenvalue Problem," *Lin. Alg. and Its Applic. 58*, 43–68.

A. Bunse-Gerstner, R. Byers, and V. Mehrmann (1992). "A Chart of Numerical Methods for Structured Eigenvalue Problems," *SIAM J. Matrix Anal. Appl. 13*, 419–453.

A. Bunse-Gerstner, R. Byers, and V. Mehrmann (1993). "Numerical Methods for Simultaneous Diagonalization," *SIAM J. Matrix Anal. Appl. 14*, 927–949.

A. Bunse-Gerstner and W.B. Gragg (1988). "Singular Value Decompositions of Complex Symmetric Matrices," *J. Comp. Applic. Math. 21*, 41–54.

J.V. Burke and M.L. Overton (1992). "Stable Perturbations of Nonsymmetric Matrices," *Lin.Alg. and Its Applic. 171*, 249–273.

P.A. Businger (1968). "Matrices Which Can be Optimally Scaled," *Numer. Math. 12*, 346–48.

P.A. Businger (1969). "Reducing a Matrix to Hessenberg Form," *Math. Comp. 23*, 819–21.

P.A. Businger (1971). "Monitoring the Numerical Stability of Gaussian Elimination," *Numer. Math. 16*, 360–61.

P.A. Businger (1971). "Numerically Stable Deflation of Hessenberg and Symmetric Tridiagonal Matrices,*BIT 11*, 262–70.

P.A. Businger and G.H. Golub (1965). "Linear Least Squares Solutions by Householder Transformations," *Numer. Math. 7*, 269–76. See also Wilkinson and Reinsch (1971,111–18).

P.A. Businger and G.H. Golub (1969). "Algorithm 358: Singular Value Decomposition of a Complex Matrix," *Comm. Assoc. Comp. Mach. 12*, 564–65.

B.L. Buzbee (1986) "A Strategy for Vectorization," *Parallel Computing 3*, 187-192.

B.L. Buzbee and F.W. Dorr (1974). "The Direct Solution of the Biharmonic Equation on Rectangular Regions and the Poisson Equation on Irregular Regions,"*SIAM J. Num. Anal. 11*, 753 63.

B.L. Buzbee, F.W. Dorr, J.A. George, and G.H. Golub (1971). "The Direct Solution of the Discrete Poisson Equation on Irregular Regions,"*SIAM J. Num. Anal. 8*, 722–36.

B.L. Buzbee, G.H. Golub, and C.W. Nielson (1970). "On Direct Methods for Solving Poisson's Equations," *SIAM J. Num. Anal. 7*, 627-56.

R. Byers (1984). "A Linpack-Style Condition Estimator for the Equation $AX - XB^T = C$," *IEEE Trans. Auto. Cont.* *AC-29*, 926–928.

R. Byers (1986) "A Hamiltonian QR Algorithm," *SIAM J. Sci. and Stat. Comp.* *7*, 212–229.

R. Byers (1988). "A Bisection Method for Measuring the Distance of a Stable Matrix to the Unstable Matrices," *SIAM J. Sci. Stat. Comp.* *9*, 875–881.

R. Byers and S.G. Nash (1987). "On the Singular Vectors of the Lyapunov Operator," *SIAM J. Alg. and Disc. Methods 8*, 59–66.

X.-C. Cai and O. Widlund (1993). "Multiplicative Schwarz Algorithms for Some Nonsymmetric and Indefinite Problems," *SIAM J. Numer. Anal.* *30*, 936–952.

D. Calvetti, G.H. Golub, and L. Reichel (1994). "An Adaptive Chebyshev Iterative Method for Nonsymmetric Linear Systems Based on Modified Moments," *Numer. Math.* *67*, 21–40.

D. Calvetti and L. Reichel (1992). "A Chebychev-Vandermonde Solver," *Lin. Alg. and Its Applic.* *172*, 219–229.

D. Calvetti and L. Reichel (1993). "Fast Inversion of Vandermonde-Like Matrices Involving Orthogonal Polynomials," *BIT 33*, 473–484.

D. Calvetti, L. Reichel, and D.C. Sorensen (1994). "An Implicitly Restarted Lanczos Method for Large Symmetric Eigenvalue Problems," *ETNA 2*, 1–21.

L.E. Cannon (1969). *A Cellular Computer to Implement the Kalman Filter Algorithm*, Ph.D. Thesis, Montana State University.

R. Carter (1991). "Y-MP Floating Point and Cholesky Factorization," *Int'l J. High Speed Computing 3*, 215–222.

F. Chaitin-Chatelin and V. Frayseé (1996). *Lectures on Finite Precision Computations*, SIAM Publications, Philadelphia, PA.

R.H. Chan (1989). "The Spectrum of a Family of Circulant Preconditioned Toeplitz Systems," *SIAM J. Num. Anal.* *26*, 503–506.

R.H. Chan (1991). "Preconditioners for Toeplitz Systems with Nonnegative Generating Functions," *IMA J. Num. Anal.* *11*, 333–345.

R.H. Chan, J.G. Nagy, and R.J. Plemmons (1993). "FFT based Preconditioners for Toeplitz Block Least Squares Problems," *SIAM J. Num. Anal.* *30*, 1740–1768.

R.H. Chan, J.G. Nagy, and R.J. Plemmons (1994). "Circulant Preconditioned Toeplitz Least Squares Iterations," *SIAM J. Matrix Anal. Appl.* *15*, 80–97.

S.P. Chan and B.N. Parlett (1977). "Algorithm 517: A Program for Computing the Condition Numbers of Matrix Eigenvalues Without Computing Eigenvectors," *ACM Trans. Math. Soft.* *3*, 186–203.

T.F. Chan (1982). "An Improved Algorithm for Computing the Singular Value Decomposition," *ACM Trans. Math. Soft.* *8*, 72–83.

T.F. Chan (1984). "Deflated Decomposition Solutions of Nearly Singular Systems," *SIAM J. Num. Anal.* *21*, 738–754.

T.F. Chan (1985). "On the Existence and Computation of LU Factorizations with small pivots," *Math. Comp.* *42*, 535–548.

T.F. Chan (1987). "Rank Revealing QR Factorizations," *Lin. Alg. and Its Applic.* *88/89*, 67–82.

T.F. Chan (1988). "An Optimal Circulant Preconditioner for Toeplitz Systems," *SIAM. J. Sci. Stat. Comp.* *9*, 766–771.

T.F. Chan (1991). "Fourier Analysis of Relaxed Incomplete Factorization Preconditioners," *SIAM J. Sci. Statist. Comput.* *12*, 668–680.

T.F. Chan and P. Hansen (1992). "A Look-Ahead Levinson Algorithm for Indefinite Toeplitz Systems," *SIAM J. Matrix Anal. Appl.* *13*, 490–506.

T.F. Chan and P. Hansen (1992). "Some Applications of the Rank Revealing QR Factorization," *SIAM J. Sci. and Stat. Comp.* *13*, 727–741.

T.F. Chan, K.R. Jackson, and B. Zhu (1983). "Alternating Direction Incomplete Factorizations," *SIAM J. Numer. Anal.* *20*, 239–257.

T.F. Chan and J.A. Olkin (1994). "Circulant Preconditioners for Toeplitz Block Matrices," *Numerical Algorithms 6*, 89–101.

T.F. Chan, J.A. Olkin, and D. Cooley (1992). "Solving Quadratically Constrained Least Squares Using Black Box Solvers," *BIT 32*, 481–495.
S. Chandrasekaren and I.C.F. Ipsen (1994). "On Rank-Revealing Factorizations," *SIAM J. Matrix Anal. Appl. 15*, 592–622.
S. Chandrasekaren and I.C.F. Ipsen (1994). "Backward Errors for Eigenvalue and Singular Value Decompositions," *Numer. Math. 68*, 215–223.
S. Chandrasekaren and I.C.F. Ipsen (1995). "On the Sensitivity of Solution Components in Linear Systems of Equations," *SIAM J. Matrix Anal. Appl. 16*, 93–112.
H.Y. Chang and M. Salama (1988). "A Parallel Householder Tridiagonalization Strategy Using Scattered Square Decomposition," *Parallel Computing 6*, 297-312.
J.P. Charlier, M. Vanbegin, P. Van Dooren (1988). "On Efficient Implementation of Kogbetliantz's Algorithm for Computing the Singular Value Decomposition," *Numer. Math. 52*, 279–300.
J.P. Charlier and P. Van Dooren (1987). "On Kogbetliantz's SVD Algorithm in the Presence of Clusters," *Lin. Alg. and Its Applic. 95*, 135–160.
B.A. Chartres and J.C. Geuder (1967). "Computable Error Bounds for Direct Solution of Linear Equations," *J. ACM 14*, 63–71.
F. Chatelin (1993). *Eigenvalues of Matrices*, John Wiley and Sons, New York.
S. Chen, J. Dongarra, and C. Hsuing (1984). "Multiprocessing Linear Algebra Algorithms on the Cray X-MP-2: Experiences with Small Granularity," *J. Parallel and Distributed Computing 1*, 22–31.
S. Chen, D. Kuck, and A. Sameh (1978). "Practical Parallel Band Triangular Systems Solvers," *ACM Trans. Math. Soft. 4*, 270–277.
K.H. Cheng and S. Sahni (1987). "VLSI Systems for Band Matrix Multiplication," *Parallel Computing 4*, 239–258.
R.C. Chin, T.A. Manteuffel, and J. de Pillis (1984). "ADI as a Preconditioning for Solving the Convection-Diffusion Equation," *SIAM J. Sci. and Stat. Comp. 5*, 281–299.
J. Choi, J.J. Dongarra, and D.W. Walker (1995). "Parallel Matrix Transpose Algorithms on Distributed Memory Concurrent Computers," *Parallel Computing 21*, 1387–1406.
M.T. Chu (1992). "Numerical Methods for Inverse Singular Value Problems," *SIAM J. Num. Anal. 29*, 885–903.
M.T. Chu, R.E. Funderlic, and G.H. Golub (1995). "A Rank-One Reduction Formula and Its Applications to Matrix Factorizations," *SIAM Review 37*, 512–530.
P.G. Ciarlet (1989). *Introduction to Numerical Linear Algebra and Optimisation*, Cambridge University Press.
A.K. Cline (1973). "An Elimination Method for the Solution of Linear Least Squares Problems," *SIAM J. Num. Anal. 10*, 283–89.
A.K. Cline (1976). "A Descent Method for the Uniform Solution to Overdetermined Systems of Equations," *SIAM J. Num. Anal. 13*, 293–309.
A.K. Cline, A.R. Conn, and C. Van Loan (1982). "Generalizing the LINPACK Condition Estimator," in *Numerical Analysis* , ed., J.P. Hennart, Lecture Notes in Mathematics no. 909, Springer-Verlag, New York.
A.K. Cline, G.H. Golub, and G.W. Platzman (1976). "Calculation of Normal Modes of Oceans Using a Lanczos Method," in *Sparse Matrix Computations* , ed. J.R. Bunch and D.J. Rose, Academic Press, New York, pp. 409–26.
A.K. Cline, C.B. Moler, G.W. Stewart, and J.H. Wilkinson (1979). "An Estimate for the Condition Number of a Matrix," *SIAM J. Num. Anal.* 16, 368-75.
A.K. Cline and R.K. Rew (1983). "A Set of Counter examples to Three Condition Number Estimators," *SIAM J. Sci. and Stat. Comp. 4*, 602–611.
R.E. Cline and R.J. Plemmons (1976). "L_2-Solutions to Underdetermined Linear Systems," *SIAM Review 18*, 92-106.
M. Clint and A. Jennings (1970). "The Evaluation of Eigenvalues and Eigenvectors of Real Symmetric Matrices by Simultaneous Iteration," *Comp. J. 13*, 76–80.
M. Clint and A. Jennings (1971). "A Simultaneous Iteration Method for the Unsymmetric Eigenvalue Problem," *J. Inst. Math. Applic. 8*, 111–21.

W.G. Cochrane (1968). "Errors of Measurement in Statistics," *Technometrics 10*, 637–66.

W.J. Cody (1988). "ALGORITHM 665 MACHAR: A Subroutine to Dynamically Determine Machine Parameters," *ACM Trans. Math. Soft. 14*, 303–311.

A.M. Cohen (1974). "A Note on Pivot Size in Gaussian Elimination," *Lin. Alg. and Its Applic. 8*, 361–68.

T.F. Coleman and Y. Li (1992). "A Globally and Quadratically Convergent Affine Scaling Method for Linear L_1 Problems," *Mathematical Programming, 56, Series A*, 189–222.

T.F. Coleman and D.C. Sorensen (1984). "A Note on the Computation of an Orthonormal Basis for the Null Space of a Matrix," *Mathematical Programming 29*, 234–242.

T.F. Coleman and C.F. Van Loan (1988). *Handbook for Matrix Computations*, SIAM Publications, Philadelphia, PA.

L. Colombet, Ph. Michallon, and D. Trystram (1996). "Parallel Matrix-Vector Product on Rings with a Minimum of Communication," *Parallel Computing 22*, 289–310.

P. Concus and G.H. Golub (1973). "Use of Fast Direct Methods for the Efficient Numerical Solution of Nonseparable Elliptic Equations," *SIAM J. Num. Anal. 10*, 1103–20.

P. Concus, G.H. Golub, and G. Meurant (1985). "Block Preconditioning for the Conjugate Gradient Method," *SIAM J. Sci. and Stat. Comp. 6*, 220–252.

P. Concus, G.H. Golub, and D.P. O'Leary (1976). " A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations," in *Sparse Matrix Computations* , ed. J.R. Bunch and D.J. Rose, Academic Press, New York.

K. Connolly, J.J. Dongarra, D. Sorensen, and J. Patterson (1988). "Programming Methodology and Performance Issues for Advanced Computer Architectures," *Parallel Computing 5*, 41-58.

J.M. Conroy (1989). "A Note on the Parallel Cholesky Factorization of Wide Banded Matrices," *Parallel Computing 10*, 239–246.

S.D. Conte and C. de Boor (1980). *Elementary Numerical Analysis: An Algorithmic Approach*, 3rd ed., McGraw-Hill, New York.

J.E. Cope and B.W. Rust (1979). "Bounds on solutions of systems with accurate data," *SIAM J. Num. Anal. 16*, 950–63.

M. Costnard, M. Marrakchi, and Y. Robert (1988). "Parallel Gaussian Elimination on an MIMD Computer," *Parallel Computing 6*, 275–296.

M. Costnard, J.M. Muller, and Y. Robert (1986). "Parallel QR Decomposition of a Rectangular Matrix," *Numer. Math. 48*, 239–250.

M. Costnard and D. Trystram (1995). *Parallel Algorithms and Architectures*, International Thomson Computer Press, New York.

R.W. Cottle (1974). "Manifestations of the Schur Complement," *Lin. Alg. and Its Applic. 8*, 189-211.

M.G. Cox (1981). "The Least Squares Solution of Overdetermined Linear Equations having Band or Augmented Band Structure," *IMA J. Num. Anal. 1*, 3–22.

C.R. Crawford (1973). "Reduction of a Band Symmetric Generalized Eigenvalue Problem," *Comm. ACM 16*, 41–44.

C.R. Crawford (1976). "A Stable Generalized Eigenvalue Problem," *SIAM J. Num. Anal. 13*, 854–60.

C.R. Crawford (1986). "Algorithm 646 PDFIND: A Routine to Find a Positive Definite Linear Combination of Two Real Symmetric Matrices," *ACM Trans. Math. Soft. 12*, 278–282.

C.R. Crawford and Y.S. Moon (1983). "Finding a Positive Definite Linear Combination of Two Hermitian Matrices," *Lin. Alg. and Its Applic. 51*, 37–48.

S. Crivelli and E.R. Jessup (1995). "The Cost of Eigenvalue Computation on Distributed Memory MIMD Computers," *Parallel Computing 21*, 401–422.

C.W. Cryer (1968). "Pivot Size in Gaussian Elimination," *Numer. Math. 12*, 335–45.

J. Cullum (1978). "The Simultaneous Computation of a Few of the Algebraically Largest and Smallest Eigenvalues of a Large Sparse Symmetric Matrix," *BIT 18*, 265–75.

J. Cullum and W.E. Donath (1974). "A Block Lanczos Algorithm for Computing the q Algebraically Largest Eigenvalues and a Corresponding Eigenspace of Large Sparse Real Symmetric Matrices," *Proc. of the 1974 IEEE Conf. on Decision and Control*, Phoenix, Arizona, pp. 505–9.

J. Cullum and R.A. Willoughby (1977). "The Equivalence of the Lanczos and the Conjugate Gradient Algorithms," IBM Research Report RE-6903.

J. Cullum and R.A. Willoughby (1979). "Lanczos and the Computation in Specified Intervals of the Spectrum of Large, Sparse Real Symmetric Matrices, in *Sparse Matrix Proc.* , 1978, ed. I.S. Duff and G.W. Stewart, SIAM Publications, Philadelphia, PA.

J. Cullum and R. Willoughby (1980). "The Lanczos Phenomena: An Interpretation Based on Conjugate Gradient Optimization," *Lin. Alg. and Its Applic. 29*, 63 90.

J. Cullum and R.A. Willoughby (1985). *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. I Theory*, Birkhaüser, Boston.

J. Cullum and R.A. Willoughby (1985). *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. II Programs*, Birkhaüser, Boston.

J. Cullum and R.A. Willoughby, eds. (1986). *Large Scale Eigenvalue Problems*, North-Holland, Amsterdam.

J. Cullum, R.A. Willoughby, and M. Lake (1983). "A Lanczos Algorithm for Computing Singular Values and Vectors of Large Matrices," *SIAM J. Sci. and Stat. Comp. 4*, 197–215.

J.J.M. Cuppen (1981). "A Divide and Conquer Method for the Symmetric Eigenproblem," *Numer. Math. 36*, 177–95.

J.J.M. Cuppen (1983). "The Singular Value Decomposition in Product Form," *SIAM J. Sci. and Stat. Comp. 4*, 216–222.

J.J.M. Cuppen (1984). "On Updating Triangular Products of Householder Matrices," *Numer. Math. 45*, 403-410.

E. Cuthill (1972). "Several Strategies for Reducing the Bandwidth of Matrices," in *Sparse Matrices and Their Applications*, ed. D.J. Rose and R.A. Willoughby, Plenum Press, New York.

G. Cybenko (1978). "Error Analysis of Some Signal Processing Algorithms," Ph.D. thesis, Princeton University.

G. Cybenko (1980). "The Numerical Stability of the Levinson-Durbin Algorithm for Toeplitz Systems of Equations," *SIAM J. Sci. and Stat. Comp. 1*, 303-19.

G. Cybenko (1984). "The Numerical Stability of the Lattice Algorithm for Least Squares Linear Prediction Problems," *BIT 24*, 441–455.

G. Cybenko (1985). "Computing Pisarenko Frequency Estimates," in *Proceedings of the Princeton Conference on Information Science and Systems*, Dept. of Electrical Engineering, Princeton University.

G. Cybenko and M. Berry (1990). "Hyperbolic Householder Algorithms for Factoring Structured Matrices," *SIAM J. Matrix Anal. Appl. 11*, 499–520.

G. Cybenko and C. Van Loan (1986). "Computing the Minimum Eigenvalue of a Symmetric Positive Definite Toeplitz Matrix," *SIAM J. Sci. and Stat. Comp. 7*, 123–131.

K. Dackland, E. Elmroth, and B. Kågström (1992). "Parallel Block Factorizations on the Shared Memory Multiprocessor IBM 3090 VF/600J," *International J. Supercomputer Applications, 6*, 69–97.

J. Daniel, W.B. Gragg, L.Kaufman, and G.W. Stewart (1976). "Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization," *Math. Comp. 30*, 772–795.

B. Danloy (1976). "On the Choice of Signs for Householder Matrices," *J. Comp. Appl. Math. 2*, 67–69.

B.N. Datta (1989). "Parallel and Large-Scale Matrix Computations in Control: Some Ideas," *Lin. Alg. and Its Applic. 121*, 243–264.

B.N. Datta (1995). *Numerical Linear Algebra and Applications*. Brooks/Cole Publishing Company, Pacific Grove, California.

B.N. Datta, C.R. Johnson. M.A. Kaashoek, R. Plemmons, and E.D. Sontag, eds. (1988), *Linear Algebra in Signals, Systems, and Control*, SIAM Publications, Philadelphia, PA.

K. Datta (1988). "The Matrix Equation $XA - BX = R$ and Its Applications," *Lin. Alg. and Its Applic. 109*, 91–105.

C. Davis and W.M. Kahan (1970). "The Rotation of Eigenvectors by a Perturbation, III," *SIAM J. Num. Anal. 7*, 1–46.

D. Davis (1973). "Explicit Functional Calculus," *Lin. Alg. and Its Applic. 6*, 193–99.

G.J. Davis (1986). "Column LU Pivoting on a Hypercube Multiprocessor," *SIAM J. Alg. and Disc. Methods* 7, 538–550.

J. Day and B. Peterson (1988). "Growth in Gaussian Elimination," *Amer. Math. Monthly 95*, 489–513.

A. Dax (1990). "The Convergence of Linear Stationary Iterative Processes for Solving Singular Unstructured Systems of Linear Equations," *SIAM Review 32*, 611–635.

C. de Boor (1979). "Efficient Computer Manipulation of Tensor Products," *ACM Trans. Math. Soft. 5*, 173–182.

C. de Boor and A. Pinkus (1977). "A Backward Error Analysis for Totally Positive Linear Systems," *Numer. Math. 27*, 485-90.

T. Dehn, M. Eiermann, K. Giebermann, and V. Sperling (1995). "Structured Sparse Matrix Vector Multiplication on Massively Parallel SIMD Architectures," *Parallel Computing 21*, 1867–1894.

P. Deift, J. Demmel, L.-C. Li, and C. Tomei (1991). "The Bidiagonal Singular Value Decomposition and Hamiltonian Mechanics," *SIAM J. Num. Anal. 28*, 1463–1516.

P. Deift, T. Nanda, and C. Tomei (1983). "Ordinary Differential Equations and the Symmetric Eigenvalue Problem," *SIAM J. Numer. Anal. 20*, 1–22.

T. Dekker and W. Hoffman (1989). "Rehabilitation of the Gauss-Jordan Algorithm," *Numer. Math. 54*, 591–599.

T.J. Dekker and J.F. Traub (1971). "The Shifted QR Algorithm for Hermitian Matrices," *Lin. Alg. and Its Applic. 4*, 137–54.

J.M. Delosme and I.C.F. Ipsen (1986). "Parallel Solution of Symmetric Positive Definite Systems with Hyperbolic Rotations," *Lin. Alg. and Its Applic. 77*, 75–112.

C.J. Demeure (1989). "Fast QR Factorization of Vandermonde Matrices," *Lin. Alg. and Its Applic. 122/123/124*, 165–194.

J.W. Demmel (1983). "A Numerical Analyst's Jordan Canonical Form," Ph.D. Thesis, Berkeley.

J.W. Demmel (1983). "The Condition Number of Equivalence Transformations that Block Diagonalize Matrix Pencils," *SIAM J. Numer. Anal. 20*, 599–610.

J.W. Demmel (1984). "Underflow and the Reliability of Numerical Software," *SIAM J. Sci. and Stat. Comp. 5*, 887–919.

J.W. Demmel (1987). "Three Methods for Refining Estimates of Invariant Subspaces," *Computing 38*, 43–57.

J.W. Demmel (1987). "On the Distance to the Nearest Ill-Posed Problem," *Numer. Math. 51*, 251–289.

J.W. Demmel (1987). "A Counterexample for two Conjectures About Stability," *IEEE Trans. Auto. Cont. AC-32*, 340–342.

J.W. Demmel (1987). "The smallest perturbation of a submatrix which lowers the rank and constrained total least squares problems, *SIAM J. Numer. Anal. 24*, 199–206.

J.W. Demmel (1988). "The Probability that a Numerical Analysis Problem is Difficult," *Math. Comp. 50*, 449–480.

J.W. Demmel (1992). "The Componentwise Distance to the Nearest Singular Matrix," *SIAM J. Matrix Anal. Appl. 13*, 10–19.

J.W. Demmel (1996). *Numerical Linear Algebra*, SIAM Publications, Philadelphia, PA.

J.W. Demmel and W. Gragg (1993). "On Computing Accurate Singular Values and Eigenvalues of Matrices with Acyclic Graphs," *Lin. Alg. and Its Applic. 185*, 203–217.

J.W. Demmel, M.T. Heath, and H.A. Van Der Vorst (1993) "Parallel Numerical Linear Algebra," in *Acta Numerica 1993*, Cambridge University Press.

J.W. Demmel and N.J. Higham (1992). "Stability of Block Algorithms with Fast Level-3 BLAS," *ACM Trans. Math. Soft. 18*, 274–291.

J.W. Demmel and N.J. Higham (1993). "Improved Error Bounds for Underdetermined System Solvers," *SIAM J. Matrix Anal. Appl. 14*, 1 14.

J.W. Demmel, N.J. Higham, and R.S. Schreiber (1995). "Stability of Block LU Factorization," *Numer. Lin. Alg. with Applic. 2*, 173–190.

J.W. Demmel and B. Kågström (1986). "Stably Computing the Kronecker Structure and Reducing Subspaces of Singular Pencils $A - \lambda B$ for Uncertain Data," in *Large Scale Eigenvalue Problems*, J. Cullum and R.A. Willoughby (eds), North-Holland, Amsterdam.

J.W. Demmel and B. Kågström (1987). "Computing Stable Eigendecompositions of Matrix Pencils," *Linear Alg. and Its Applic 88/89*, 139–186.

J.W. Demmel and B. Kågström (1988). "Accurate Solutions of Ill-Posed Problems in Control Theory," *SIAM J. Matrix Anal. Appl.* 126–145.

J.W. Demmel and W. Kahan (1990). "Accurate Singular Values of Bidiagonal Matrices," *SIAM J. Sci. and Stat. Comp. 11*, 873–912.

J.W. Demmel and K. Veselič (1992). "Jacobi's Method is More Accurate than QR," *SIAM J. Matrix Anal. Appl. 13*, 1204-1245.

B. De Moor and G.H. Golub (1991). "The Restricted Singular Value Decomposition: Properties and Applications," *SIAM J. Matrix Anal. Appl. 12*, 401–425.

B. De Moor and P. Van Dooren (1992). "Generalizing the Singular Value and QR Decompositions," *SIAM J. Matrix Anal. Appl. 13*, 993 1014.

J.E. Dennis and R.B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.

J.E. Dennis Jr. and K. Turner (1987). "Generalized Conjugate Directions," *Lin. Alg. and Its Applic. 88/89*, 187–209.

E.F. Deprettere, ed. (1988). *SVD and Signal Processing*. Elsevier, Amsterdam.

J. Descloux (1963). "Bounds for the Spectral Norm of Functions of Matrices," *Numer. Math. 5*, 185–90.

M.A. Diamond and D.L.V. Ferreira (1976). "On a Cyclic Reduction Method for the Solution of Poisson's Equation," *SIAM J. Num. Anal. 13*, 54–70.

S. Doi (1991). "On Parallelism and Convergence of Incomplete LU Factorizations," *Appl. Numer. Math. 7*, 417–436.

J.J. Dongarra (1983). "Improving the Accuracy of Computed Singular Values," *SIAM J. Sci. and Stat. Comp. 4*, 712–719.

J.J. Dongarra, J.R. Bunch, C.B. Moler, and G.W. Stewart (1979). *LINPACK Users Guide*, SIAM Publications, Philadelphia, PA.

J.J. Dongarra, J. Du Croz, I.S. Duff, and S.J. Hammarling (1990). "A Set of Level 3 Basic Linear Algebra Subprograms," *ACM Trans. Math. Soft. 16*, 1–17.

J.J. Dongarra, J. Du Croz, I.S. Duff, and S.J. Hammarling (1990). "Algorithm 679. A Set of Level 3 Basic Linear Algebra Subprograms: Model Implementation and Test Programs," *ACM Trans. Math. Soft. 16*, 18–28.

J.J. Dongarra, J. Du Croz, S. Hammarling, and R.J. Hanson (1988). "An Extended Set of Fortran Basic Linear Algebra Subprograms," *ACM Trans. Math. Soft. 14*, 1–17.

J.J. Dongarra, J. Du Croz, S. Hammarling, and R.J. Hanson (1988). "Algorithm 656 An Extended Set of Fortran Basic Linear Algebra Subprograms: Model Implementation and Test Programs," *ACM Trans. Math. Soft. 14*, 18–32.

J.J. Dongarra, I. Duff, P. Gaffney, and S. McKee, eds. (1989), *Vector and Parallel Computing*, Ellis Horwood, Chichester, England.

J.J. Dongarra, I. Duff, D. Sorensen, and H. van der Vorst (1990). *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM Publications, Philadelphia, PA.

J.J. Dongarra and S. Eisenstat (1984). "Squeezing the Most Out of an Algorithm in Cray Fortran," *ACM Trans. Math. Soft. 10*, 221-230.

J.J. Dongarra, F.G. Gustavson, and A. Karp (1984). "Implementing Linear Algebra Algorithms for Dense Matrices on a Vector Pipeline Machine," *SIAM Review 26*, 91–112.

J.J. Dongarra, S. Hammarling, and D.C. Sorensen (1989). "Block Reduction of Matrices to Condensed Forms for Eigenvalue Computations," *JACM 27*, 215–227.

J.J. Dongarra, S. Hammarling, and J.H. Wilkinson (1992). "Numerical Considerations in Computing Invariant Subspaces," *SIAM J. Matrix Anal. Appl. 13*, 145–161.

J.J. Dongarra and T. Hewitt (1986). "Implementing Dense Linear Algebra Algorithms Using Multitasking on the Cray X-MP-4 (or Approaching the Gigaflop)," *SIAM J. Sci. and Stat. Comp. 7*, 347–350.

J.J. Dongarra and A. Hinds (1979). "Unrolling Loops in Fortran," *Software Practice and Experience 9*, 219-229.

J.J. Dongarra and R.E. Hiromoto (1984). "A Collection of Parallel Linear Equation Routines for the Denelcor HEP,' *Parallel Computing 1*, 133–142.

J.J. Dongarra, L. Kaufman, and S. Hammarling (1986). "Squeezing the Most Out of Eigenvalue Solvers on High Performance Computers," *Lin. Alg. and Its Applic. 77*, 113–136.

J.J. Dongarra, C.B. Moler, and J.H. Wilkinson (1983). "Improving the Accuracy of Computed Eigenvalues and Eigenvectors," *SIAM J. Numer. Anal. 20*, 23–46.

J.J. Dongarra and A.H. Sameh (1984). "On Some Parallel Banded System Solvers," *Parallel Computing 1*, 223–235.

J.J. Dongarra, A. Sameh, and D. Sorensen (1986). "Implementation of Some Concurrent Algorithms for Matrix Factorization," *Parallel Computing 3*, 25–34.

J.J. Dongarra and D.C. Sorensen (1986). "Linear Algebra on High Performance Computers," *Appl. Math. and Comp. 20*, 57–88.

J.J. Dongarra and D.C. Sorensen (1987). "A Portable Environment for Developing Parallel Programs," *Parallel Computing 5*, 175–186.

J.J. Dongarra and D.C. Sorensen (1987). "A Fully Parallel Algorithm for the Symmetric Eigenvalue Problem," *SIAM J. Sci. and Stat. Comp. 8*, S139–S154.

J.J. Dongarra and D. Walker (1995). "Software Libraries for Linear Algebra Computations on High Performance Computers," *SIAM Review 37*, 151–180.

F.W. Dorr (1970). "The Direct Solution of the Discrete Poisson Equation on a Rectangle," *SIAM Review 12*, 248–63.

F.W. Dorr (1973). "The Direct Solution of the Discrete Poisson Equation in $O(n^2)$ Operations," *SIAM Review 15*, 412–415.

C.C. Douglas, M. Heroux, G. Slishman, and R.M. Smith (1994). "GEMMW: A Portable Level 3 BLAS Winograd Variant of Strassen's Matrix-Matrix Multiply Algorithm," *J. Comput. Phys. 110*, 1–10.

Z. Drmač (1994). *The Generalized Singular Value Problem*, Ph.D. Thesis, FernUniversitat, Hagen, Germany.

Z. Dramăc, M. Omladič, and K. Veselič (1994). "On the Perturbation of the Cholesky Factorization," *SIAM J. Matrix Anal. Appl. 15*,1319–1332.

P.F. Dubois, A. Greenbaum, and G.H. Rodrigue (1979). "Approximating the Inverse of a Matrix for Use on Iterative Algorithms on Vector Processors," *Computing 22*, 257–268.

A.A. Dubrulle (1970). "A Short Note on the Implicit QL Algorithm for Symmetric Tridiagonal Matrices," *Numer. Math. 15*, 450.

A.A. Dubrulle and G.H. Golub (1994). "A Multishift QR Iteration Without Computation of the Shifts," *Numerical Algorithms 7*, 173–181.

A.A. Dubrulle, R.S. Martin, and J.H. Wilkinson (1968). "The Implicit QL Algorithm," *Numer. Math. 12*, 377-83. see also Wilkinson and Reinsch (1971, pp.241–48).

J.J. Du Croz and N.J. Higham (1992). "Stability of Methods for Matrix Inversion," *IMA J. Num. Anal. 12*, 1–19.

I.S. Duff (1974). "Pivot Selection and Row Ordering in Givens Reduction on Sparse Matrices," *Computing 13*, 239–48.

I.S. Duff (1977). "A Survey of Sparse Matrix Research," *Proc. IEEE 65*, 500–535.

I.S. Duff, ed. (1981). *Sparse Matrices and Their Uses*, Academic Press, New York.

I.S. Duff, A.M. Erisman, and J.K. Reid (1986). *Direct Methods for Sparse Matrices*, Oxford University Press.

I.S. Duff, N.I.M. Gould, J.K. Reid, J.A. Scott, and K. Turner (1991). "The Factorization of Sparse Indefinite Matrices," *IMA J. Num. Anal. 11*, 181–204.

I.S. Duff and G. Meurant (1989). "The Effect of Ordering on Preconditioned Conjugate Gradients," *BIT 29*, 635–657.

I.S. Duff and J.K. Reid (1975). "On the Reduction of Sparse Matrices to Condensed Forms by Similarity Transformations," *J. Inst. Math. Applic. 15*, 217–24.

I.S. Duff and J.K. Reid (1976). "A Comparison of Some Methods for the Solution of Sparse Over-Determined Systems of Linear Equations," *J. Inst. Math. Applic. 17*, 267–80.

I.S. Duff and G.W. Stewart, eds. (1979). *Sparse Matrix Proceedings, 1978*, SIAM Publications, Philadelphia, PA.

N. Dunford and J. Schwartz (1958). *Linear Operators, Part I*, Interscience, New York.

J. Durbin (1960). "The Fitting of Time Series Models," *Rev. Inst. Int. Stat. 28* 233–43.

P.J. Eberlein (1965). "On Measures of Non-Normality for Matrices," *Amer. Math. Soc. Monthly 72*, 995–96.

P.J. Eberlein (1970). "Solution to the Complex Eigenproblem by a Norm-Reducing Jacobi-type Method," *Numer. Math. 14*, 232–45. See also Wilkinson and Reinsch (1971, pp.404–17).

P.J. Eberlein (1971). "On the Diagonalization of Complex Symmetric Matrices," *J. Inst. Math. Applic. 7*, 377–83.

P.J. Eberlein (1987). "On Using the Jacobi Method on a Hypercube," in *Hypercube Multiprocessors*, ed. M.T. Heath, SIAM Publications, Philadelphia, PA.

P.J. Eberlein and C.P. Huang (1975). "Global Convergence of the QR Algorithm for Unitary Matrices with Some Results for Normal Matrices," *SIAM J. Numer. Anal. 12*, 421–453.

C. Eckart and G. Young (1939). "A Principal Axis Transformation for Non-Hermitian Matrices," *Bull. Amer. Math. Soc. 45*, 118–21.

A. Edelman (1992). "The Complete Pivoting Conjecture for Gaussian Elimination is False," *The Mathematica Journal 2*, 58–61.

A. Edelman (1993). "Large Dense Numerical Linear Algebra in 1993: The Parallel Computing Influence," *Int'l J. Supercomputer Appl. 7*, 113–128.

A. Edelman, E. Elmroth, and B. Kågström (1996). "A Geometric Approach to Perturbation Theory of Matrices and Matrix Pencils," *SIAM J. Matrix Anal.*, to appear.

A. Edelman and W. Mascarenhas (1995). "On the Complete Pivoting Conjecture for a Hadamard Matrix of Order 12," *Linear and Multilinear Algebra 38*, 181 185.

A. Edelman and H. Murakami (1995). "Polynomial Roots from Companion Matrix Eigenvalues," *Math. Comp. 64*, 763–776.

M. Eiermann and W. Niethammer (1983). "On the Construction of Semi-iterative Methods," *SIAM J. Numer. Anal. 20*, 1153–1160.

M. Eiermann, W. Niethammer, and R.S. Varga (1992). "Acceleration of Relaxation Methods for Non-Hermitian Linear Systems," *SIAM J. Matrix Anal. Appl. 13*, 979–991.

M. Eiermann and R.S. Varga (1993). "Is the Optimal ω Best for the SOR Iteration Method," *Lin. Alg. and Its Applic. 182*, 257–277.

V. Eijkhout (1991). "Analysis of Parallel Incomplete Point Factorizations," *Lin. Alg. and Its Applic. 154–156*, 723–740.

S.C. Eisenstat (1984). "Efficient Implementation of a Class of Preconditioned Conjugate Gradient Methods," *SIAM J. Sci. and Stat. Computing 2*, 1 4.

S.C. Eisenstat, H. Elman, and M. Schultz (1983). "Variational Iterative Methods for Nonsymmetric Systems of Equations," *SIAM J. Num. Anal. 20*, 345–357.

S.C. Eisenstat, M.T. Heath, C.S. Henkel, and C.H. Romine (1988). "Modified Cyclic Algorithms for Solving Triangular Systems on Distributed Memory Multiprocessors," *SIAM J. Sci. and Stat. Comp. 9*, 589–600.

L. Eldén (1977). "Algorithms for the Regularization of Ill-Conditioned Least Squares Problems," *BIT 17*, 134–45.

L. Eldén (1980). "Perturbation Theory for the Least Squares Problem with Linear Equality Constraints," *SIAM J. Num. Anal. 17*, 338–50.

L. Eldén (1983). "A Weighted Pseudoinverse, Generalized Singular Values, and Constrained Least Squares Problems," *BIT 22* , 487–502.

L. Eldén (1984). "An Algorithm for the Regularization of Ill-Conditioned, Banded Least Squares Problems," *SIAM J. Sci. and Stat. Comp. 5*, 237–254.

L. Eldén (1985). "A Note on the Computation of the Generalized Cross-Validation Function for Ill-Conditioned Least Squares Problems," *BIT 24*, 467–472.

L. Eldén and H. Park (1994). "Perturbation Analysis for Block Downdating of a Cholesky Decomposition," *Numer. Math. 68*, 457–468.

L. Eldén and H. Park (1994). "Block Downdating of Least Squares Solutions," *SIAM J. Matrix Anal. Appl. 15*, 1018–1034.

L. Eldén and R. Schreiber (1986). "An Application of Systolic Arrays to Linear Discrete Ill-Posed Problems," *SIAM J. Sci. and Stat. Comp. 7*, 892–903.

H. Elman (1986). "A Stability Analysis of Incomplete LU Factorization," *Math. Comp. 47*, 191-218.

H. Elman (1989). "Approximate Schur Complement Preconditioners on Serial and Parallel Computers," *SIAM J. Sci. Stat. Comput. 10*, 581–605.

H. Elman (1996). "Fast Nonsymmetric Iterations and Preconditioning for Navier-Stokes Equations," *SIAM J. Sci. Comput. 17*, 33–46.

H. Elman and G.H. Golub (1990). "Iterative Methods for Cyclically Reduced Non-Self-Adjoint Linear Systems I," *Math. Comp. 54*, 671–700.

H. Elman and G.H. Golub (1990). "Iterative Methods for Cyclically Reduced Non-Self-Adjoint Linear Systems II," *Math. Comp. 56*, 215–242.

E. Elmroth and B. Kågström (1996). "The Set of 2-by-3 Matrix Pencils–Kronecker Structure and their Transitions under Perturbations," *SIAM J. Matrix Anal.*, to appear.

L. Elsner and J.-G. Sun (1982). "Perturbation Theorems for the Generalized Eigenvalue Problem,; *Lin. Alg. and its Applic. 48*, 341-357.

W. Enright (1979). "On the Efficient and Reliable Numerical Solution of Large Linear Systems of O.D.E.'s," *IEEE Trans. Auto. Cont.* AC-24, 905–8.

W. Enright and S. Serbin (1978). "A Note on the Efficient Solution of Matrix Pencil Systems," *BIT 18*, 276–81.

I. Erdelyi (1967). "On the Matrix Equation $Ax = \lambda Bx$," *J. Math. Anal. and Applic. 17*, 119–32.

T. Ericsson and A. Ruhe (1980). "The Spectral Transformation Lanczos Method for the Numerical Solution of Large Sparse Generalized Symmetric Eigenvalue Problems," *Math. Comp. 35*, 1251–68.

A.M. Erisman and J.K. Reid (1974). "Monitoring the Stability of the Triangular Factorization of a Sparse Matrix," *Numer. Math. 22*, 183–86.

J. Erxiong (1990). "An Algorithm for Finding Generalized Eigenpairs of a Symmetric Definite Matrix Pencil," *Lin.Alg. and Its Applic. 132*, 65–91.

J. Erxiong (1992). "A Note on the Double-Shift QL Algorithm," *Lin.Alg. and Its Applic. 171*, 121–132.

D.J. Evans (1984). "Parallel SOR Iterative Methods," *Parallel Computing 1*, 3–18.

D.J. Evans and R. Dunbar (1983). "The Parallel Solution of Triangular Systems of Equations," *IEEE Trans. Comp. C-32*, 201–204.

L.M. Ewerbring and F.T. Luk (1989). "Canonical Correlations and Generalized SVD: Applications and New Algorithms," *J. Comput. Appl. Math. 27*, 37–52.

V. Faber and T. Manteuffel (1984). "Necessary and Sufficient Conditions for the Existence of a Conjugate Gradient Method," *SIAM J. Numer. Anal. 21* 352–362.

V.N. Faddeeva (1959). *Computational Methods of Linear Algebra*, Dover, New York.

V. Fadeeva and D. Fadeev (1977). "Parallel Computations in Linear Algebra," *Kibernetica 6*, 28 40.

W. Fair and Y. Luke (1970). "Padé Approximations to the Operator Exponential," *Numer. Math. 14*, 379–82.

R.W. Farebrother (1987). *Linear Least Squares Computations*, Marcel Dekker, New York.

D.G. Feingold and R.S. Varga (1962). "Block Diagonally Dominant Matrices and Generalizations of the Gershgorin Circle Theorem," *Pacific J. Math. 12*, 1241–50.

T. Fenner and G. Loizou (1974). "Some New Bounds on the Condition Numbers of Optimally Scaled Matrices," *J. ACM 21*, 514–24.

W.E. Ferguson (1980). "The Construction of Jacobi and Periodic Jacobi Matrices with Prescribed Spectra," *Math. Comp. 35*, 1203–1220.

K.V. Fernando (1989). "Linear Convergence of the Row Cyclic Jacobi and Kogbetliantz methods," *Numer. Math. 56*, 73–92.

K.V. Fernando and B.N. Parlett (1994). "Accurate Singular Values and Differential qd Algorithms," *Numer. Math. 67*, 191–230.

W.R. Ferng, G.H. Golub, and R.J. Plemmons (1991). "Adaptive Lanczos Methods for Recursive Condition Estimation," *Numerical Algorithms 1*, 1-20.

R.D. Fierro and J.R. Bunch (1994). "Collinearity and Total Least Squares," *SIAM J. Matrix Anal. Appl. 15*, 1167–1181.

R.D. Fierro and P.C. Hansen (1995). "Accuracy of TSVD Solutions Computed from Rank-Revealing Decompositions," *Numer. Math. 70*, 453–472.

C. Fischer and R.A. Usmani (1969). "Properties of Some Tridiagonal Matrices and Their Application to Boundary Value Problems," *SIAM J. Num. Anal. 6*, 127–42.

G. Fix and R. Heiberger (1972). "An Algorithm for the Ill-Conditioned Generalized Eigenvalue Problem," *SIAM J. Num. Anal. 9*, 78–88.

U. Flaschka, W-W. Li, and J-L. Wu (1992). "A KQZ Algorithm for Solving Linear-Response Eigenvalue Equations," *Lin. Alg. and Its Applic. 165*, 93–123.

R. Fletcher (1976). "Factorizing Symmetric Indefinite Matrices," *Lin. Alg. and Its Applic. 14*, 257–72.

A. Forsgren (1995). "On Linear Least-Squares Problems with Diagonally Dominant Weight Matrices," Technical Report TRITA-MAT-1995-OS2, Department of Mathematics, Royal Institute of Technology, S-100 44, Stockholm, Sweden.

G.E. Forsythe (1960). "Crout with Pivoting," *Comm. ACM 3*, 507–8.

G.E. Forsythe and G.H. Golub (1965). "On the Stationary Values of a Second-Degree Polynomial on the Unit Sphere," *SIAM J. App. Math. 13*, 1050–68.

G.E. Forsythe and P. Henrici (1960). "The Cyclic Jacobi Method for Computing the Principal Values of a Complex Matrix," *Trans. Amer. Math. Soc. 94*, 1–23.

G.E. Forsythe and C. Moler (1967). *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs, NJ.

L.V. Foster (1986). "Rank and Null Space Calculations Using Matrix Decomposition without Column Interchanges," *Lin. Alg. and Its Applic. 74*, 47–71.

L.V. Foster (1994). "Gaussian Elimination with Partial Pivoting Can Fail in Practice," *SIAM J. Matrix Anal. Appl. 15*, 1354–1362.

R. Fourer (1984). "Staircase Matrices and Systems," *SIAM Review 26*, 1–71.

L. Fox (1964). *An Introduction to Numerical Linear Algebra*, Oxford University Press, Oxford, England.

G.C. Fox, ed. (1988). *The Third Conference on Hypercube Concurrent Computers and Applications, Vol. II – Applications*, ACM Press, New York.

G.C. Fox, M. A. Johnson, G. A. Lyzenga, S. W. Otto, J. K. Salmon and D. W. Walker (1988). *Solving Problems on Concurrent Processors, Volume 1.* Prentice Hall, Englewood Cliffs, NJ.

G.C. Fox, S.W. Otto, and A.J. Hey (1987). "Matrix Algorithms on a Hypercube I: Matrix Multiplication," *Parallel Computing 4*, 17 31.

G.C. Fox, R.D. Williams, and P. C. Messina (1994). *Parallel Computing Works!*, Morgan Kaufmann, San Francisco.

J.S. Frame (1964). "Matrix Functions and Applications, Part II," *IEEE Spectrum 1 (April)*, 102–8.

J.S. Frame (1964). "Matrix Functions and Applications, Part IV," *IEEE Spectrum 1 (June),* 123–31.

J.G.F. Francis (1961). "The QR Transformation: A Unitary Analogue to the LR Transformation, Parts I and II" *Comp. J. 4*, 265–72, 332–45.

J.N. Franklin (1968). *Matrix Theory* Prentice Hall, Englewood Cliffs, NJ.

T. L. Freeman and C. Phillips (1992). *Parallel Numerical Algorithms*, Prentice Hall, New York.

R.W. Freund (1990). "On Conjugate Gradient Type Methods and Polynomial Preconditioners for a Class of Complex Non-Hermitian Matrices," *Numer. Math. 57*, 285–312.

R.W. Freund (1992). "Conjugate Gradient-Type Methods for Linear Systems with Complex Symmetric Coefficient Matrices," *SIAM J. Sci. Statist. Comput. 13*, 425–448.

R.W. Freund (1993). "A Transpose-Free Quasi-Minimum Residual Algorithm for Non-hermitian Linear System," *SIAM J. Sci. Comput. 14*, 470–482.

R.W. Freund and N. Nachtigal (1991). "QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems," *Numer. Math. 60*, 315–339.

R.W. Freund and N.M. Nachtigal (1994). "An Implementation of the QMR Method Based on Coupled Two-term Recurrences," *SIAM J. Sci. Comp. 15*, 313–337.

R.W. Freund, G.H. Golub, and N. Nachtigal (1992). "Iterative Solution of Linear Systems," *Acta Numerica 1*, 57–100.

R.W. Freund, M. Gutknecht, and N. Nachtigal (1993). "An Implementation of the Look-Ahead Lanczos Algorithm for Non-Hermitian Matrices," *SIAM J. Sci. and Stat.Comp. 14*, 137–158.

R.W. Freund and H. Zha (1993). "A Look-Ahead Algorithm for the Solution of General Hankel Systems," *Numer. Math. 64*, 295–322.

S. Friedland (1975). "On Inverse Multiplicative Eigenvalue Problems for Matrices," *Lin. Alg. and Its Applic. 12,* 127–38.

S. Friedland (1991). "Revisiting Matrix Squaring," *Lin. Alg. and Its Applic. 154-156,* 59–63.

S. Friedland, J. Nocedal, and M.L. Overton (1987). "The Formulation and Analysis of Numerical Methods for Inverse Eigenvalue Problems," *SIAM J. Numer. Anal. 24,* 634–667.

C.E. Froberg (1965). "On Triangularization of Complex Matrices by Two Dimensional Unitary Tranformations," *BIT 5,* 230–34.

R.E. Funderlic and A. Geist (1986). "Torus Data Flow for Parallel Computation of Missized Matrix Problems," *Lin. Alg. and Its Applic. 77,* 149–164.

G. Galimberti and V. Pereyra (1970). "Numerical Differentiation and the Solution of Multidimensional Vandermonde Systems," *Math. Comp. 24,* 357–64.

G. Galimberti and V. Pereyra (1971). "Solving Confluent Vandermonde Systems of Hermitian Type," *Numer. Math. 18,* 44–60.

K.A Gallivan, M. Heath, E. Ng, J. Ortega, B. Peyton, R. Plemmons, C. Romine, A. Sameh, and B. Voigt (1990), *Parallel Algorithms for Matrix Computations*, SIAM Publications, Philadelphia, PA.

K.A. Gallivan, W. Jalby, and U. Meier (1987). "The Use of BLAS3 in Linear Algebra on a Parallel Processor with a Hierarchical Memory," *SIAM J. Sci. and Stat. Comp. 8,* 1079-1084.

K.A. Gallivan, W. Jalby, U. Meier, and A.H. Sameh (1988). "Impact of Hierarchical Memory Systems on Linear Algebra Algorithm Design," *Int'l J. Supercomputer Applic. 2,* 12-48.

K.A. Gallivan, R.J. Plemmons, and A.H. Sameh (1990). "Parallel Algorithms for Dense Linear Algebra Computations," *SIAM Review 32*, 54–135.

E. Gallopoulos and Y. Saad (1989). "A Parallel Block Cyclic Reduction Algorithm for the Fast Solution of Elliptic Equations," *Parallel Computing 10*, 143–160.

W. Gander (1981). "Least Squares with a Quadratic Constraint," *Numer. Math. 36,* 291-307.

W. Gander, G.H. Golub, and U. von Matt (1991). "A Constrained Eigenvalue Problem," in *Numerical Linear Algebra, Digital Signal Processing, and Parallel Algorithms*, G.H. Golub and P. Van Dooren (eds), Springer-Verlag, Berlin.

D. Gannon and J. Van Rosendale (1984). "On the Impact of Communication Complexity on the Design of Parallel Numerical Algorithms," *IEEE Trans. Comp. C-33*, 1180–1194.

F.R. Gantmacher (1959). *The Theory of Matrices, vols. 1 and 2*, Chelsea, New York.

B.S. Garbow, J.M. Boyle, J.J. Dongarra, and C.B. Moler (1972). *Matrix Eigensystem Routines: EISPACK Guide Extension*, Lecture Notes in Computer Science, Volume 51, Springer-Verlag, New York.

J. Gardiner, M.R. Wette, A.J. Laub, J.J. Amato, and C.B. Moler (1992). "Algorithm 705: A FORTRAN-77 Software Package for Solving the Sylvester Matrix Equation $AXB^T + CXD^T = E$," *ACM Trans. Math. Soft. 18*, 232–238.

W. Gautschi (1975). "Norm Estimates for Inverses of Vandermonde Matrices," *Numer. Math. 23*, 337–47.

W. Gautschi (1975). "Optimally Conditioned Vandermonde Matrices," *Numer. Math. 24*, 1–12.

G.A. Geist (1991). "Reduction of a General Matrix to Tridiagonal Form," *SIAM J. Matrix Anal. Appl. 12*, 362–373.

G.A. Geist and M.T. Heath (1986). "Matrix Factorization on a Hypercube," in M.T. Heath (ed) (1986). *Proceedings of First SIAM Conference on Hypercube Multiprocessors*, SIAM Publications, Philadelphia, PA.

G.A. Geist and C.H. Romine (1988). "LU Factorization Algorithms on Distributed Memory Multiprocessor Architectures," *SIAM J. Sci. and Stat. Comp. 9*, 639–649.

W.M. Gentleman (1973). "Least Squares Computations by Givens Transformations without Square Roots," *J. Inst. Math. Appl. 12*, 329–36.

W.M. Gentleman (1973). "Error Analysis of QR Decompositions by Givens Transformations," *Lin. Alg. and Its Applic. 10*, 189–97.

W.M. Gentleman and H.T. Kung (1981). "Matrix Triangularization by Systolic Arrays," SPIE Proceedings, Vol. 298, 19–26.

J.A. George (1973). "Nested Dissection of a Regular Finite Element Mesh," *SIAM J. Num. Anal. 10*, 345–63.

J.A. George (1974). "On Block Elimination for Sparse Linear Systems," *SIAM J. Num. Anal. 11*, 585–603.

J.A. George and M.T. Heath (1980). "Solution of Sparse Linear Least Squares Problems Using Givens Rotations," *Lin. Alg. and Its Applic. 34*, 69–83.

A. George, M.T. Heath, and J. Liu (1986). "Parallel Cholesky Factorization on a Shared Memory Multiprocessor," *Lin. Alg. and Its Applic. 77*, 165–187.

A. George and J. W-H. Liu (1981). *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall Inc., Englewood Cliffs, New Jersey.

A.R. Ghavimi and A.J. Laub (1995). "Residual Bounds for Discrete-Time Lyapunov Equations," *IEEE Trans. Auto. Cont. 40*, 1244–1249.

N.E. Gibbs and W.G. Poole, Jr. (1974). "Tridiagonalization by Permutations," *Comm. ACM 17*, 20–24.

N.E. Gibbs, W.G. Poole, Jr., and P.K. Stockmeyer (1976). "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix," *SIAM J. Num. Anal. 13*, 236–50.

N.E. Gibbs, W.G. Poole, Jr., and P.K. Stockmeyer (1976). "A Comparison of Several Bandwidth and Profile Reduction Algorithms," *ACM Trans. Math. Soft. 2*, 322–30.

P.E. Gill, G.H. Golub, W. Murray, and M.A. Saunders (1974). "Methods for Modifying Matrix Factorizations," *Math. Comp. 28*, 505-35.

P.E. Gill and W. Murray (1976). "The Orthogonal Factorization of a Large Sparse Matrix," in *Sparse Matrix Computations*, ed. J.R. Bunch and D.J. Rose, Academic Press, New York, pp. 177-200.

P.E. Gill, W. Murray, D.B. Ponceleón, and M.A. Saunders (1992). "Preconditioners for Indefinite Systems Arising in Optimization," *SIAM J. Matrix Anal. Appl. 13*, 292–311.

P.E. Gill, W. Murray, and M.A. Saunders (1975). "Methods for Computing and Modifying the LDV Factors of a Matrix," *Math. Comp.* *29*, 1051–77.

P.E. Gill, W. Murray, and M.H. Wright (1991). *Numerical Linear Algebra and Optimization, Vol. 1*, Addison-Wesley, Reading, MA.

W. Givens (1958). "Computation of Plane Unitary Rotations Transforming a General Matrix to Triangular Form," *SIAM J. App. Math.* *6*, 26–50.

J. Gluchowska and A. Smoktunowicz (1990). "Solving the Linear Least Squares Problem with Very High Relative Accuracy," *Computing 45*, 345–354.

I.C. Gohberg and M.G. Krein (1969). *Introduction to the Theory of Linear Non-Self Adjoint Operators* , Amer. Math. Soc., Providence, R.I.

I.C. Gohberg, P. Lancaster, and L. Rodman (1986). *Invariant Subspaces of Matrices With Applications*, John Wiley and Sons, New York.

D. Goldberg (1991). "What Every Computer Scientist Should Know About Floating Point Arithmetic," *ACM Surveys 23*, 5–48.

D. Goldfarb (1976). "Factored Variable Metric Methods for Unconstrained Optimization," *Math. Comp. 30*, 796–811.

H.H. Goldstine and L.P. Horowitz (1959). "A Procedure for the Diagonalization of Normal Matrices," *J. Assoc. Comp. Mach. 6*, 176–95.

G.H. Golub (1965). "Numerical Methods for Solving Linear Least Squares Problems," *Numer. Math. 7*, 206–16.

G.H. Golub (1969). "Matrix Decompositions and Statistical Computation," in *Statistical Computation* , ed. R.C. Milton and J.A. Nelder, Academic Press, New York, pp. 365–97.

G.H. Golub (1973). "Some Modified Matrix Eigenvalue Problems," *SIAM Review 15*, 318–334.

G.H. Golub (1974). "Some Uses of the Lanczos Algorithm in Numerical Linear Algebra," *in Topics in Numerical Analysis*, ed., J.J.H. Miller, Academic Press, New York.

G.H. Golub, M. Heath, and G. Wahba (1979). "Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter," *Technometrics 21*, 215–23.

G.H. Golub, A. Hoffman, and G.W. Stewart (1988). "A Generalization of the Eckart-Young-Mirsky Approximation Theorem." *Lin. Alg. and Its Applic. 88/89*, 317–328.

G.H. Golub and W. Kahan (1965). "Calculating the Singular Values and Pseudo-Inverse of a Matrix," *SIAM J. Num. Anal. 2*, 205–24.

G.H. Golub, V. Klema and G.W. Stewart (1976). "Rank Degeneracy and Least Squares Problems," Technical Report TR-456, Department of Computer Science, University of Maryland, College Park, MD.

G.H. Golub, F.T. Luk, and M. Overton (1981). "A Block Lanczos Method for Computing the Singular Values and Corresponding Singular Vectors of a Matrix," *ACM Trans. Math. Soft. 7*, 149–69.

G.H. Golub and G. Meurant (1983). *Résolution Numérique des Grandes Systèmes Linéaires,* Collection de la Direction des Etudes et Recherches de l'Electricité de France, vol. 49, Eyolles, Paris.

G.H. Golub and C.D. Meyer (1986). "Using the QR Factorization and Group Inversion to Compute, Differentiate, and estimate the Sensitivity of Stationary Probabilities for Markov Chains," *SIAM J. Alg. and Dis. Methods*, 7, 273–281.

G.H. Golub, S. Nash, and C. Van Loan (1979). "A Hessenberg-Schur Method for the Matrix Problem $AX + XB = C$," *IEEE Trans. Auto. Cont. AC-24*, 909–13.

G.H. Golub and D. O'Leary (1989). "Some History of the Conjugate Gradient and Lanczos Methods," *SIAM Review 31*, 50–102.

G.H. Golub and J.M. Ortega (1993). *Scientific Computing: An Introduction with Parallel Computing,* Academic Press, Boston.

G.H. Golub and M. Overton (1988). "The Convergence of Inexact Chebychev and Richardson Iterative Methods for Solving Linear Systems," *Numer. Math. 53*, 571–594.

G.H. Golub and V. Pereyra (1973). "The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate," *SIAM J. Num. Anal. 10*, 413–32.

G.H. Golub and V. Pereyra (1976). "Differentiation of Pseudo-Inverses, Separable Nonlinear Least Squares Problems and Other Tales," in *Generalized Inverses and Applications* , ed. M.Z. Nashed, Academic Press, New York, pp. 303–24.

G.H. Golub and C. Reinsch (1970). "Singular Value Decomposition and Least Squares Solutions," *Numer. Math. 14*, 403–20. See also Wilkinson and Reinsch (1971, 134–51).

G.H. Golub and W.P. Tang (1981). "The Block Decomposition of a Vandermonde Matrix and Its Applications,"*BIT 21*, 505–17.

G.H. Golub and R. Underwood (1977). "The Block Lanczos Method for Computing Eigenvalues," in *Mathematical Software III* , ed. J. Rice, Academic Press, New York, pp. 364–77.

G.H. Golub, R. Underwood, and J.H. Wilkinson (1972). "The Lanczos Algorithm for the Symmetric $Ax = \lambda Bx$ Problem," Report STAN-CS-72-270, Department of Computer Science, Stanford University, Stanford, California.

G.H. Golub and P. Van Dooren, eds. (1991). *Numerical Linear Algebra, Digital Signal Processing, and Parallel Algorithms.*. Springer-Verlag, Berlin.

G.H. Golub and C.F. Van Loan (1979). "Unsymmetric Positive Definite Linear Systems," *Lin. Alg. and Its Applic. 28*, 85–98.

G.H. Golub and C.F. Van Loan (1980). "An Analysis of the Total Least Squares Problem," *SIAM J. Num. Anal. 17*, 883–93.

G.H. Golub and J.M. Varah (1974). "On a Characterization of the Best L_2-Scaling of a Matrix," *SIAM J. Num. Anal. 11*, 472–79.

G.H. Golub and R.S. Varga (1961). "Chebychev Semi-Iterative Methods, Successive Over-Relaxation Iterative Methods, and Second-Order Richardson Iterative Methods, Parts I and II," *Numer. Math. 3*, 147 56, 157–68.

G.H. Golub and J.H. Welsch (1969). "Calculation of Gauss Quadrature Rules," *Math. Comp. 23*, 221–30.

G.H. Golub and J.H. Wilkinson (1966). "Note on Iterative Refinement of Least Squares Solutions," *Numer. Math. 9*, 139–48.

G.H. Golub and J.H. Wilkinson (1976). "Ill-Conditioned Eigensystems and the Computation of the Jordan Canonical Form," *SIAM Review 18*, 578–619.

G.H. Golub and H. Zha (1994). "Perturbation Analysis of the Canonical Correlations of Matrix Pairs," *Lin. Alg. and Its Applic. 210*, 3–28.

N. Gould (1991). "On Growth in Gaussian Elimination with Complete Pivoting," *SIAM J. Matrix Anal. Appl. 12*, 354–361.

R.J. Goult, R.F. Hoskins, J.A. Milner and M.J. Pratt (1974). *Computational Methods in Linear Algebra*, John Wiley and Sons, New York.

A.R. Gourlay (1970). "Generalization of Elementary Hermitian Matrices," *Comp. J. 13*, 411–12.

A.R. Gourlay and G.A. Watson (1973). *Computational Methods for Matrix Eigenproblems*, John Wiley & Sons, New York.

W. Govaerts (1991). "Stable Solvers and Block Elimination for Bordered Systems," *SIAM J. Matrix Anal. Appl. 12*, 469–483.

W. Govaerts and J.D. Pryce (1990). "Block Elimination with One Iterative Refinement Solves Bordered Linear Systems Accurately," *BIT 30*, 490–507.

W. Govaerts and J.D. Pryce (1993). "Mixed Block Elimination for Linear Systems with Wider Borders," *IMA J. Num. Anal. 13*, 161–180.

W. B. Gragg (1986). "The QR Algorithm for Unitary Hessenberg Matrices," *J. Comp. Appl. Math. 16*, 1–8.

W.B. Gragg and W.J. Harrod (1984). "The Numerically Stable Reconstruction of Jacobi Matrices from Spectral Data," *Numer. Math. 44*, 317–336.

W.B. Gragg and L. Reichel (1990). "A Divide and Conquer Method for Unitary and Orthogonal Eigenproblems," *Numer. Math. 57*, 695–718.

A. Graham (1981). *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood Ltd., Chichester, England.

B. Green (1952). "The Orthogonal Approximation of an Oblique Structure in Factor Analysis," *Psychometrika 17*, 429–40.

A. Greenbaum (1992). "Diagonal Scalings of the Laplacian as Preconditioners for Other Elliptic Differential Operators," *SIAM J. Matrix Anal. Appl. 13*, 826–846.

A. Greenbaum and G. Rodrigue (1989). "Optimal Preconditioners of a Given Sparsity Pattern," *BIT 29*, 610–634.

A. Greenbaum and Z. Strakos (1992). "Predicting the Behavior of Finite Precision Lanczos and Conjugate Gradient Computations," *SIAM J. Matrix Anal. Appl. 13*, 121 137.

A. Greenbaum and L.N. Trefethen (1994). "GMRES/CR and Arnoldi/Lanczos as Matrix Approximation Problems," *SIAM J. Sci. Comp. 15*, 359–368.

J. Greenstadt (1955). "A Method for Finding Roots of Arbitrary Matrices," *Math. Tables and Other Aids to Comp. 9*, 47–52.

R.G. Grimes and J.G. Lewis (1981). "Condition Number Estimation for Sparse Matrices," *SIAM J. Sci. and Stat. Comp. 2*, 384–88.

R.G. Grimes, J.G. Lewis, and H.D. Simon (1994). "A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems," *SIAM J. Matrix Anal. Appl. 15*, 228–272.

W.D. Gropp and D.E. Keyes (1988). "Complexity of Parallel Implementation of Domain Decomposition Techniques for Elliptic Partial Differential Equations," *SIAM J. Sci. and Stat. Comp. 9*, 312–326.

W.D. Gropp and D.E. Keyes (1992). "Domain Decomposition with Local Mesh Refinement," *SIAM J. Sci. Statist. Comput. 13*, 967–993.

M. Gu and S.C. Eisenstat (1995). "A Divide-and-Conquer Algorithm for the Bidiagonal SVD," *SIAM J. Matrix Anal. Appl. 16*, 79–92.

M. Gu and S.C. Eisenstat (1995). "A Divide-and-Conquer Algorithm for the Symmetric Tridiagonal Eigenproblem," *SIAM J. Matrix Anal. Appl. 16*, 172–191.

M. Gulliksson (1994). "Iterative Refinement for Constrained and Weighted Linear Least Squares," *BIT 34*, 239–253.

M. Gulliksson (1995). "Backward Error Analysis for the Constrained and Weighted Linear Least Squares Problem When Using the Weighted QR Factorization," *SIAM J. Matrix. Anal. Appl. 13*, 675–687.

M. Gulliksson and P-Å. Wedin (1992). "Modifying the QR-Decomposition to Constrained and Weighted Linear Least Squares," *SIAM J. Matrix Anal. Appl. 13*, 1298–1313.

R.F. Gunst, J.T. Webster, and R.L. Mason (1976). "A Comparison of Least Squares and Latent Root Regression Estimators," *Technometrics 18*, 75–83.

K.K. Gupta (1972). "Solution of Eigenvalue Problems by Sturm Sequence Method," *Int. J. Numer. Meth. Eng. 4*, 379–404.

M. Gutknecht (1992). "A Completed Theory of the Unsymmetric Lanczos Process and Related Algorithms, Part I," *SIAM J. Matrix Anal. Appl. 13*, 594–639.

M. Gutknecht (1993). "Variants of BiCBSTAB for Matrices with Complex Spectrum," *SIAM J. Sci. and Stat. Comp. 14*, 1020–1033.

M. Gutknecht (1994). "A Completed Theory of the Unsymmetric Lanczos Process and Related Algorithms, Part II," *SIAM J. Matrix Anal. Appl. 15*, 15–58.

W. Hackbusch (1994). *Iterative Solution of Large Sparse Systems of Equations*, Springer-Verlag, New York.

D. Hacon (1993). "Jacobi's Method for Skew-Symmetric Matrices," *SIAM J. Matrix Anal. Appl. 14*, 619–628.

L.A. Hageman and D.M. Young (1981). *Applied Iterative Methods*, Academic Press, New York.

W.W. Hager (1984). "Condition Estimates," *SIAM J. Sci. and Stat. Comp. 5*, 311–316.

W.W. Hager (1988). *Applied Numerical Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ.

S.J. Hammarling (1974). "A Note on Modifications to the Givens Plane Rotation," *J. Inst. Math. Appl. 13*, 215–18.

S.J. Hammarling (1985). "The Singular Value Decomposition in Multivariate Statistics," *ACM SIGNUM Newsletter 20*, 2-25.

S.L. Handy and J.L. Barlow (1994). "Numerical Solution of the Eigenproblem for Banded, Symmetric Toeplitz Matrices," *SIAM J. Matrix Anal. Appl. 15*, 205–214.

M. Hanke and J.G. Nagy (1994). "Toeplitz Approximate Inverse Preconditioner for Banded Toeplitz Matrices," *Numerical Algorithms 7*, 183–199.

M. Hanke and M. Neumann (1990). "Preconditionings and Splittings for Rectangular Systems," *Numer. Math. 57*, 85–96.

E.R. Hansen (1962). "On Quasicyclic Jacobi Methods," *ACM J. 9*, 118–35.

E.R. Hansen (1963). "On Cyclic Jacobi Methods," *SIAM J. Appl. Math. 11*, 448–59.

P.C. Hansen (1987). "The Truncated SVD as a Method for Regularization," *BIT 27*, 534–553.

P.C. Hansen (1988). "Reducing the Number of Sweeps in Hestenes Method," in *Singular Value Decomposition and Signal Processing*, ed. E.F. Deprettere, North Holland.

P.C. Hansen (1990). "Relations Between SVD and GSVD of Discrete Regularization Problems in Standard and General Form," *Lin.Alg. and Its Applic. 141*, 165–176.

P.C. Hansen and H. Gesmar (1993). "Fast Orthogonal Decomposition of Rank-Deficient Toeplitz Matrices," *Numerical Algorithms 4*, 151–166.

R.J. Hanson and C.L. Lawson (1969). "Extensions and Applications of the Householder Algorithm for Solving Linear Least Square Problems," *Math. Comp. 23*, 787 812.

V. Hari (1982). "On the Global Convergence of the Eberlein Method for Real Matrices," *Numer. Math. 39*, 361–370.

V. Hari (1991). "On Pairs of Almost Diagonal Matrices," *Lin. Alg. and Its Applic. 148*, 193–223.

M.T. Heath, ed. (1986). *Proceedings of First SIAM Conference on Hypercube Multiprocessors*, SIAM Publications, Philadelphia, PA.

M.T. Heath, ed. (1987). *Hypercube Multiprocessors*, SIAM Publications, Philadelphia, PA.

M.T. Heath (1997). *Scientific Computing: An Introductory Survey*, McGraw-Hill, New York.

M.T. Heath, A.J. Laub, C.C. Paige, and R.C. Ward (1986). "Computing the SVD of a Product of Two Matrices," *SIAM J. Sci. and Stat. Comp. 7*, 1147-1159.

M.T. Heath, E. Ng, and B.W. Peyton (1991). "Parallel Algorithms for Sparse Linear Systems," *SIAM Review 33*, 420–460.

M.T. Heath and C.H. Romine (1988). "Parallel Solution of Triangular Systems on Distributed Memory Multiprocessors," *SIAM J. Sci. and Stat. Comp. 9*, 558–588.

M. Hegland (1991). "On the Parallel Solution of Tridiagonal Systems by Wrap-Around Partitioning and Incomplete LU Factorization," *Numer. Math. 59*, 453–472.

G. Heinig and P. Jankowski (1990). "Parallel and Superfast Algorithms for Hankel Systems of Equations," *Numer. Math. 58*, 109–127.

D.E. Heller (1976). "Some Aspects of the Cyclic Reduction Algorithm for Block Tridiagonal Linear Systems,"*SIAM J. Num. Anal. 13*, 484–96.

D.E. Heller (1978). "A Survey of Parallel Algorithms in Numerical Linear Algebra," *SIAM Review 20*, 740-777.

D.E. Heller and I.C.F. Ipsen (1983). "Systolic Networks for Orthogonal Decompositions," *SIAM J. Sci. and Stat. Comp. 4*, 261–269.

B.W. Helton (1968). "Logarithms of Matrices," *Proc. Amer. Math. Soc. 19*, 733–36.

H.V. Henderson and S.R. Searle (1981). "The Vec-Permutation Matrix, The Vec Operator, and Kronecker Products: A Review," *Linear and Multilinear Algebra 9*, 271–288.

B. Hendrickson and D. Womble (1994). "The Torus-Wrap Mapping for Dense Matrix Calculations on Massively Parallel Computers," *SIAM J. Sci. Comput. 15*, 1201–1226.

C.S. Henkel, M.T. Heath, and R.J. Plemmons (1988). "Cholesky Downdating on a Hypercube," in G. Fox (1988), 1592–1598.

P. Henrici (1958). "On the Speed of Convergence of Cyclic and Quasicyclic Jacobi Methods for Computing the Eigenvalues of Hermitian Matrices," *SIAM J. Appl. Math. 6*, 144–62.

P. Henrici (1962). "Bounds for Iterates, Inverses, Spectral Variation and Fields of Values of Non-normal Matrices," *Numer. Math. 4*, 24–40.

P. Henrici and K. Zimmermann (1968). "An Estimate for the Norms of Certain Cyclic Jacobi Operators," *Lin. Alg. and Its Applic. 1*, 489–501.

M.R. Hestenes (1980). *Conjugate Direction Methods in Optimization*, Springer-Verlag, Berlin.

M.R. Hestenes (1990). "Conjugacy and Gradients," in *A History of Scientific Computing*, Addison-Wesley, Reading, MA.

M.R. Hestenes and E. Stiefel (1952). "Methods of Conjugate Gradients for Solving Linear Systems," *J. Res. Nat. Bur. Stand. 49*, 409–36.

G. Hewer and C. Kenney (1988). "The Sensitivity of the Stable Lyapunov Equation," *SIAM J. Control Optim 26*, 321 344.

D.J. Higham (1995). "Condition Numbers and Their Condition Numbers," *Lin. Alg. and Its Applic. 214*, 193–213.

D.J. Higham and N.J. Higham (1992). "Componentwise Perturbation Theory for Linear Systems with Multiple Right-Hand Sides," *Lin. Alg. and Its Applic. 174*, 111–129.

D.J. Higham and N.J. Higham (1992). "Backward Error and Condition of Structured Linear Systems," *SIAM J. Matrix Anal. Appl. 13*, 162–175.

D.J. Higham and L.N. Trefethen (1993). "Stiffness of ODES," *BIT 33*, 285–303.

N.J. Higham (1985). "Nearness Problems in Numerical Linear Algebra," PhD Thesis, University of Manchester, England.

N.J. Higham (1986). "Newton's Method for the Matrix Square Root," *Math. Comp. 46*, 537–550.

N.J. Higham (1986). "Computing the Polar Decomposition—with Applications," *SIAM J. Sci. and Stat. Comp. 7*, 1160–1174.

N.J. Higham (1986). "Efficient Algorithms for computing the condition number of a tridiagonal matrix," *SIAM J. Sci. and Stat. Comp. 7*, 150–165.

N.J. Higham (1987). "A Survey of Condition Number Estimation for Triangular Matrices," *SIAM Review 29*, 575–596.

N.J. Higham (1987). "Error Analysis of the Björck-Pereyra Algorithms for Solving Vandermonde Systems," *Numer. Math. 50*, 613–632.

N.J. Higham (1987). "Computing Real Square Roots of a Real Matrix," *Lin. Alg. and Its Applic. 88/89*, 405–430.

N.J. Higham (1988). "Fast Solution of Vandermonde-like Systems Involving Orthogonal Polynomials," *IMA J. Num. Anal. 8*, 473-486.

N.J. Higham (1988). "Computing a Nearest Symmetric Positive Semidefinite Matrix," *Lin. Alg. and Its Applic. 103*, 103–118.

N.J. Higham (1988). "The Symmetric Procrustes Problem," *BIT 28*, 133–43.

N.J. Higham (1988). "FORTRAN Codes for Estimating the One-Norm of a Real or Complex Matrix with Applications to Condition Estimation (Algorithm 674)," *ACM Trans. Math. Soft. 14*, 381–396.

N.J. Higham (1989). "Matrix Nearness Problems and Applications," in *Applications of Matrix Theory*, M.J.C. Gover and S. Barnett (eds), Oxford University Press, Oxford UK, 1–27.

N.J. Higham (1989). "The Accuracy of Solutions to Triangular Systems," *SIAM J. Num. Anal. 26*, 1252–1265.

N.J. Higham (1990). "Bounding the Error in Gaussian Elimination for Tridiagonal Systems," *SIAM J. Matrix Anal. Appl. 11*, 521–530.

N.J. Higham (1990). "Stability Analysis of Algorithms for Solving Confluent Vandermonde-like Systems," *SIAM J. Matrix Anal. Appl. 11*, 23–41.

N.J. Higham (1990). "Analysis of the Cholesky Decomposition of a Semidefinite Matrix," in *Reliable Numerical Computation*, M.G. Cox and S.J. Hammarling (eds), Oxford University Press, Oxford, UK, 161 185.

N.J. Higham (1990). "Exploiting Fast Matrix Multiplication within the Level 3 BLAS," *ACM Trans. Math. Soft. 16*, 352–368.

N.J. Higham (1991). "Iterative Refinement Enhances the Stability of QR Factorization Methods for Solving Linear Equations," *BIT 31*, 447-468.

N.J. Higham (1992). "Stability of a Method for Multiplying Complex Matrices with Three Real Matrix Multiplications," *SIAM J. Matrix Anal. Appl. 13*, 681–687.

N.J. Higham (1992). "Estimating the Matrix p-Norm," *Numer. Math. 62*, 539–556.

N.J. Higham (1993). "Optimization by Direct Search in Matrix Computations," *SIAM J. Matrix Anal. Appl. 14*, 317–333.

N.J. Higham (1993). "Perturbation Theory and Backward Error for AX - XB = C," *BIT 33*, 124–136.

N.J. Higham (1994). "The Matrix Sign Decomposition and Its Relation to the Polar Decomposition," *Lin. Alg. and Its Applic. 212/213*, 3–20.

N.J. Higham (1994). "A Survey of Componentwise Perturbation Theory in Numerical Linear Algebra," in *Mathematics of Computation 1943–1993: A Half Century of Computational Mathematics*, W. Gautschi (ed.), Volume 48 of *Proceedings of Symposia in Applied Mathematics*, American Mathematical Society, Providence, Rhode Island.

N.J. Higham (1995). "Stability of Parallel Triangular System Solvers," *SIAM J. Sci. Comp. 16*, 400–413.

N.J. Higham (1996). *Accuracy and Stability of Numerical Algorithms*, SIAM Publications, Philadelphia, PA.

N.J. Higham and D.J. Higham (1989). "Large Growth Factors in Gaussian Elimination with Pivoting," *SIAM J. Matrix Anal. Appl. 10*, 155–164.

N.J. Higham and P.A. Knight (1995). "Matrix Powers in Finite Precision Arithmetic," *SIAM J. Matrix Anal. Appl. 16*, 343–358.

N.J. Higham and P. Papadimitriou (1994). "A Parallel Algorithm for Computing the Polar Decomposition," *Parallel Comp. 20*, 1161–1173.

R.W. Hockney (1965). "A Fast Direct Solution of Poisson's Equation Using Fourier Analysis, "*J. ACM 12*, 95-113.

R.W. Hockney and C.R. Jesshope (1988). *Parallel Computers 2*, Adam Hilger, Bristol and Philadelphia.

W. Hoffman and B.N. Parlett (1978). "A New Proof of Global Convergence for the Tridiagonal QL Algorithm," *SIAM J. Num. Anal. 15*, 929–37.

S. Holmgren and K. Otto (1992). "Iterative Solution Methods and Preconditioners for Block-Tridiagonal Systems of Equations," *SIAM J. Matrix Anal. Appl. 13*, 863-886.

H. Hotelling (1957). "The Relations of the Newer Multivariate Statistical Methods to Factor Analysis," *Brit. J. Stat. Psych. 10*, 69–79.

P.D. Hough and S.A. Vavasis (1996). "Complete Orthogonal Decomposition for Weighted Least Squares," *SIAM J. Matrix Anal. Appl.*, to appear.

A.S. Householder (1958). "Unitary Triangularization of a Nonsymmetric Matrix," *J. ACM. 5*, 339–42.

A.S. Householder (1964). *The Theory of Matrices in Numerical Analysis* , Dover Publications, New York.

A.S. Householder (1968). "Moments and characteristic Roots II," *Numer. Math. 11*, 126–28.

R. Horn and C. Johnson (1985). *Matrix Analysis*, Cambridge University Press, New York.

R. Horn and C. Johnson (1991). *Topics in Matrix Analysis*, Cambridge University Press, New York.

C.P. Huang (1975). "A Jacobi-Type Method for Triangularizing an Arbitrary Matrix," *SIAM J. Num. Anal. 12*, 566–70.

C.P. Huang (1981). "On the Convergence of the QR Algorithm with Origin Shifts for Normal Matrices," *IMA J. Num. Anal. 1*, 127–33.

C.-M. Huang and D.P. O'Leary (1993). "A Krylov Multisplitting Algorithm for Solving Linear Systems of Equations," *Lin. Alg. and Its Applic. 194*, 9–29.

T. Huckle (1992). "Circulant and Skewcirculant Matrices for Solving Toeplitz Matrix Problems," *SIAM J. Matrix Anal. Appl. 13*, 767–777.

T. Huckle (1992). "A Note on Skew-Circulant Preconditioners for Elliptic Problems," *Numerical Algorithms 2*, 279–286.

T. Huckle (1994). "The Arnoldi Method for Normal Matrices," *SIAM J. Matrix Anal. Appl. 15*, 479–489.

T. Huckle (1995). "Low-Rank Modification of the Unsymmetric Lanczos Algorithm," *Math.Comp. 64*, 1577–1588.

T.E. Hull and J.R. Swensen (1966). "Tests of Probabilistic Models for Propagation of Roundoff Errors," *Comm. ACM. 9*, 108–13.

T-M. Hwang, W-W. Lin, and E.K. Yang (1992). "Rank-Revealing LU Factorizations," *Lin. Alg. and Its Applic. 175*, 115–141.

Y. Ikebe (1979). "On Inverses of Hessenberg Matrices," *Lin. Alg. and Its Applic. 24*, 93–97.

I.C.F. Ipsen, Y. Saad, and M. Schultz (1986). "Dense Linear Systems on a Ring of Processors," *Lin. Alg. and Its Applic. 77*, 205–239.

C.G.J. Jacobi (1846). "Uber ein Leichtes Verfahren Die in der Theorie der Sacularstroungen Vorkommendern Gleichungen Numerisch Aufzulosen," *Crelle's J. 30*, 51–94.

P. Jacobson, B. Kågström, and M. Rannar (1992). "Algorithm Development for Distributed Memory Multicomputers Using Conlab," *Scientific Programming, 1*, 185–203.

H.J. Jagadish and T. Kailath (1989). "A Family of New Efficient Arrays for Matrix Multiplication," *IEEE Trans. Comput. 38*, 149–155.

W. Jalby and B. Philippe (1991). "Stability Analysis and Improvement of the Block Gram-Schmidt Algorithm," *SIAM J. Sci. Stat. Comp. 12*, 1058–1073.

M. Jankowski and M. Wozniakowski (1977). "Iterative Refinement Implies Numerical Stability," *BIT 17*, 303–311.

K.C. Jea and D.M. Young (1983). "On the Simplification of Generalized Conjugate Gradient Methods for Nonsymmetrizable Linear Systems," *Lin. Alg. and Its Applic. 52/53*, 399–417.

A. Jennings (1977). "Influence of the Eigenvalue Spectrum on the Convergence Rate of the Conjugate Gradient Method," *J. Inst. Math. Applic. 20*, 61–72.

A. Jennings (1977). *Matrix Computation for Engineers and Scientists*, John Wiley and Sons, New York.

A. Jennings and J.J. McKeowen (1992). *Matrix Computation (2nd ed)*, John Wiley and Sons, New York.

A. Jennings and D.R.L. Orr (1971). "Application of the Simultaneous Iteration Method to Undamped Vibration Problems," *Inst. J. Numer. Math. Eng. 3*, 13–24.

A. Jennings and M.R. Osborne (1977). "Generalized Eigenvalue Problems for Certain Unsymmetric Band Matrices," *Lin. Alg. and Its Applic. 29*, 139–50.

A. Jennings and W.J. Stewart (1975). "Simultaneous Iteration for the Partial Eigensolution of Real Matrices," *J. Inst. Math. Applic. 15*, 351–62.

L.S. Jennings and M.R. Osborne (1974). "A Direct Error Analysis for Least Squares," *Numer. Math. 22*, 322–32.

P.S. Jenson (1972). "The Solution of Large Symmetric Eigenproblems by Sectioning," *SIAM J. Num. Anal. 9*, 534–45.

E.R. Jessup and D.C. Sorensen (1994). "A Parallel Algorithm for Computing the Singular Value Decomposition of a Matrix," *SIAM J. Matrix Anal. Appl. 15*, 530–548.

Z. Jia (1995). "The Convergence of Generalized Lanczos Methods for Large Unsymmetric Eigenproblems," *SIAM J. Matrix Anal. Applic 16*, 543–562.

J. Johnson and C.L. Phillips (1971). "An Algorithm for the Computation of the Integral of the State Transition Matrix," *IEEE Trans. Auto. Cont. AC-16*, 204–5.

O.G. Johnson, C.A. Micchelli, and G. Paul (1983). "Polynomial Preconditioners for Conjugate Gradient Calculations," *SIAM J. Numer. Anal. 20*, 362–376.

R.J. Johnston (1971). "Gershgorin Theorems for Partitioned Matrices," *Lin. Alg. and Its Applic. 4*, 205–20.

S.L. Johnsson (1985). "Solving Narrow Banded Systems on Ensemble Architectures," *ACM Trans. Math. Soft. 11,* 271–288.

S.L. Johnsson (1986). "Band Matrix System Solvers on Ensemble Architectures," in *Supercomputers: Algorithms, Architectures, and Scientific Computation,* eds. F.A. Matsen and T. Tajima, University of Texas Press, Austin TX., 196–216.

S.L. Johnsson (1987). "Communication Efficient Basic Linear Algebra Computations on Hypercube Multiprocessors," *J. Parallel and Distributed Computing,* No. 4, 133–172.

S.L. Johnsson (1987). "Solving Tridiagonal Systems on Ensemble Architectures," *SIAM J. Sci. and Stat. Comp. 8,* 354–392.

S.L. Johnsson and C.T. Ho (1988). "Matrix Transposition on Boolean n-cube Configured Ensemble Architectures," *SIAM J. Matrix Anal. Appl. 9,* 419–454.

S.L. Johnsson and W. Lichtenstein (1993). "Block Cyclic Dense Linear Algebra," *SIAM J. Sci.Comp. 14,* 1257–1286.

S.L. Johnsson and K. Mathur (1989). "Experience with the Conjugate Gradient Method for Stress Analysis on a Data Parallel Supercomputer," *International Journal on Numerical Methods in Engineering 27,* 523–546.

P. Joly and G. Meurant (1993). "Complex Conjugate Gradient Methods," *Numerical Algorithms 4,* 379–406.

M.T. Jones and M.L. Patrick (1993). "Bunch-Kaufman Factorization for Real Symmetric Indefinite Banded Matrices," *SIAM J. Matrix Anal. Appl. 14,* 553–559.

M.T. Jones and M.L. Patrick (1994). "Factoring Symmetric Indefinite Matrices on High-Performance Architectures," *SIAM J. Matrix Anal. Appl. 15,* 273–283.

T. Jordan (1984). "Conjugate Gradient Preconditioners for Vector and Parallel Processors," in G. Birkoff and A. Schoenstadt (eds), *Proceedings of the Conference on Elliptic Problem Solvers,* Academic Press, NY.

W. Joubert (1992). "Lanczos Methods for the Solution of Nonsymmetric Systems of Linear Equations," *SIAM J. Matrix Anal. Appl. 13,* 926–943.

B. Kågström (1977). "Numerical Computation of Matrix Functions," Department of Information Processing Report UMINF-58.77, University of Ümea, Sweden.

B. Kågström (1977). "Bounds and Perturbation Bounds for the Matrix Exponential," *BIT 17,* 39–57.

B. Kågström (1985). "The Generalized Singular Value Decomposition and the General $A - \lambda B$ Problem," *BIT 24,* 568–583.

B. Kågström (1986). "RGSVD: An Algorithm for Computing the Kronecker Structure and Reducing Subspaces of Singular $A - \lambda B$ Pencils," *SIAM J. Sci. and Stat. Comp. 7,*185–211.

B. Kågström (1994). "A Perturbation Analysis of the Generalized Sylvester Equation $(AR - LB, DR - LE) = (C, F)$," *SIAM J. Matrix Anal. Appl. 15,* 1045–1060.

B. Kågström, P. Ling, and C. Van Loan (1991). "High-Performance Level-3 BLAS: Sample Routines for Double Precision Real Data," in *High Performance Computing II,* M. Durand and F. El Dabaghi (eds), North-Holland, 269–281.

B. Kågström, P. Ling, and C. Van Loan (1995). "GEMM-Based Level-3 BLAS: High-Performance Model Implementations and Performance Evaluation Benchmark," in *Parallel Programming and Applications,* P. Fritzon and L. Finmo (eds), ISO Press, 184–188.

B. Kågström and P. Poromaa (1992). "Distributed and Shared Memory Block Algorithms for the Triangular Sylvester Equation with sep^{-1} Estimators," *SIAM J. Matrix Anal. Appl. 13,* 90–101.

B. Kågström and A. Ruhe (1980). "An Algorithm for Numerical Computation of the Jordan Normal Form of a Complex Matrix," *ACM Trans. Math. Soft. 6,* 398–419.

B. Kågström and A. Ruhe (1980). "Algorithm 560 JNF: An Algorithm for Numerical Computation of the Jordan Normal Form of a Complex Matrix," *ACM Trans. Math. Soft. 6,* 437–43.

B. Kågström and A. Ruhe, eds. (1983). *Matrix Pencils,* Proc. Pite Havsbad, 1982, Lecture Notes in Mathematics 973, Springer-Verlag, New York and Berlin.

B. Kågström and L. Westin (1989). "Generalized Schur Methods with Condition Estimators for Solving the Generalized Sylvester Equation," *IEEE Trans. Auto. Cont. AC-34*, 745–751.

W. Kahan (1966). "Numerical Linear Algebra," *Canadian Math. Bull. 9*, 757–801.

W. Kahan (1975). "Spectra of Nearly Hermitian Matrices," *Proc. Amer. Math. Soc. 48*, 11–17.

W. Kahan and B.N. Parlett (1976). "How Far Should You Go with the Lanczos Process?" in *Sparse Matrix Computations*, ed. J. Bunch and D. Rose, Academic Press, New York, pp. 131-44.

W. Kahan, B.N. Parlett, and E. Jiang (1982). "Residual Bounds on Approximate Eigensystems of Nonnormal Matrices," *SIAM J. Numer. Anal. 19*, 470–484.

D. Kahaner, C.B. Moler, and S. Nash (1988). *Numerical Methods and Software*, Prentice-Hall, Englewood Cliffs, NJ.

T. Kailath and J. Chun (1994). "Generalized Displacement Structure for Block-Toeplitz, Toeplitz-Block, and Toeplitz-Derived Matrices," *SIAM J. Matrix Anal. Appl. 15*, 114–128.

T. Kailath and A.H. Sayed (1995). "Displacement Structure: Theory and Applications," *SIAM Review 37*, 297–386.

C. Kamath and A. Sameh (1989). "A Projection Method for Solving Nonsymmetric Linear Systems on Multiprocessors," *Parallel Computing 9*, 291–312.

S. Kaniel (1966). "Estimates for Some Computational Techniques in Linear Algebra," *Math. Comp. 20*, 369–78.

I.E. Kaporin (1994). "New Convergence Results and Preconditioning Strategies for the Conjugate Gradient Method," *Num. Lin. Alg. Applic. 1*, 179–210.

R.N. Kapur and J.C. Browne (1984). "Techniques for Solving Block Tridiagonal Systems on Reconfigurable Array Computers," *SIAM J. Sci. and Stat. Comp. 5*, 701-719.

I. Karasalo (1974). "A Criterion for Truncation of the QR Decomposition Algorithm for the Singular Linear Least Squares Problem," *BIT 14*, 156–66.

E.M. Kasenally (1995). "GMBACK: A Generalized Minimum Backward Error Algorithm for Nonsymmetric Linear Systems," *SIAM J. Sci. Comp. 16*, 698–719.

T. Kato (1966). *Perturbation Theory for Linear Operators*, Springer-Verlag, New York.

L. Kaufman (1974). "The LZ Algorithm to Solve the Generalized Eigenvalue Problem," *SIAM J. Num. Anal. 11*, 997–1024.

L. Kaufman (1977). "Some Thoughts on the QZ Algorithm for Solving the Generalized Eigenvalue Problem," *ACM Trans. Math. Soft. 3*, 65–75.

L. Kaufman (1979). "Application of Dense Householder Transformations to a Sparse Matrix," *ACM Trans. Math. Soft. 5*, 442–51.

L. Kaufman (1987). "The Generalized Householder Transformation and Sparse Matrices," *Lin. Alg. and Its Applic. 90*, 221–234.

L. Kaufman (1993). "An Algorithm for the Banded Symmetric Generalized Matrix Eigenvalue Problem," *SIAM J. Matrix Anal. Appl. 14*, 372–389.

J. Kautsky and G.H. Golub (1983). "On the Calculation of Jacobi Matrices," *Lin. Alg. and Its Applic. 52/53*, 439–456.

C.S. Kenney and A.J. Laub (1989). "Condition Estimates for Matrix Functions," *SIAM J. Matrix Anal. Appl. 10*, 191–209.

C.S. Kenney and A.J. Laub (1991). "Rational Iterative Methods for the Matrix Sign Function," *SIAM J. Matrix Anal. Appl. 12*, 273–291.

C.S. Kenney and A.J. Laub (1992). "On Scaling Newton's Method for Polar Decomposition and the Matrix Sign Function," *SIAM J. Matrix Anal. Appl. 13*, 688–706.

C.S. Kenney and A.J. Laub (1994). "Small-Sample Statistical Condition Estimates for General Matrix Functions," *SIAM J. Sci. Comp. 15*, 36–61.

D. Kershaw(1982). "Solution of Single Tridiagonal Linear Systems and Vectorization of the ICCG Algorithm on the Cray-1," in G. Roderigue (ed), *Parallel Computation*, Academic Press, NY, 1982.

D.E. Keyes, T.F. Chan, G. Meurant, J.S. Scroggs, and R.G. Voigt (eds) (1992). *Domain Decomposition Methods for Partial Differential Equations*, SIAM Publications, Philadelphia, PA.

A. Kielbasinski (1987). "A Note on Rounding Error Analysis of Cholesky Factorization," *Lin. Alg. and Its Applic. 88/89*, 487–494.

S.K. Kim and A.T. Chronopoulos (1991). "A Class of Lanczos-Like Algorithms Implemented on Parallel Computers," *Parallel Comput. 17*, 763–778.

F. Kittaneh (1995). "Singular Values of Companion Matrices and Bounds on Zeros of Polynomials," *SIAM J. Matrix Anal. Appl. 16*, 333–340.

P.A. Knight (1993). "Error Analysis of Stationary Iteration and Associated Problems," Ph.D. thesis, Department of Mathematics, University of Manchester, England.

P.A. Knight (1995). "Fast Rectangular Matrix Multiplication and the QR Decomposition," *Lin. Alg. and Its Applic. 221*, 69–81.

D. Knuth (1981). *The Art of Computer Programming , vol. 2. Seminumerical Algorithms* , 2nd ed., Addison-Wesley, Reading, Massachusetts.

E.G. Kogbetliantz (1955). "Solution of Linear Equations by Diagonalization of Coefficient Matrix," *Quart. Appl. Math. 13*, 123–132.

S. Kourouklis and C.C. Paige (1981). "A Constrained Least Squares Approach to the General Gauss-Markov Linear Model," *J. Amer. Stat. Assoc. 76*, 620–25.

V.N. Kublanovskaya (1961). "On Some Algorithms for the Solution of the Complete Eigenvalue Problem," *USSR Comp. Math. Phys. 3*, 637–57.

V.N. Kublanovskaya (1984). "AB Algorithm and Its Modifications for the Spectral Problem of Linear Pencils of Matrices," *Numer. Math. 43*, 329–342.

V.N. Kublanovskaja and V.N. Fadeeva (1964). "Computational Methods for the Solution of a Generalized Eigenvalue Problem," *Amer. Math. Soc. Transl. 2*, 271–90.

J. Kuczyński and H. Woźniakowski (1992). "Estimating the Largest Eigenvalue by the Power and Lanczos Algorithms with a Random Start," *SIAM J. Matrix Anal. Appl. 13*, 1094–1122.

U.W. Kulisch and W.L. Miranker (1986). "The Arithmetic of the Digital Computer," *SIAM Review 28*, 1–40.

V. Kumar, A. Grama, A. Gupta and G. Karypis (1994). *Introduction to Parallel Computing: Design and Analysis of Algorithms*, Benjamin/Cummings, Reading, MA.

H.T. Kung (1982). "Why Systolic Architectures?," *Computer 15*, 37-46.

C.D. La Budde (1964). "Two Classes of Algorithms for Finding the Eigenvalues and Eigenvectors of Real Symmetric Matrices," *J. ACM 11*, 53–58.

S. Lakshmivarahan and S. K. Dhall (1990). *Analysis and Design of Parallel Algorithms: Arithmetic and Matrix Problems*, McGraw-Hill, New York.

J. Lambiotte and R.G. Voigt (1975). "The Solution of Tridiagonal Linear Systems of the CDC-STAR 100 Computer," *ACM Trans. Math. Soft. 1*, 308–29.

P. Lancaster (1970). "Explicit Solution of Linear Matrix Equations," *SIAM Review 12*, 544–66.

P. Lancaster and M. Tismenetsky (1985). *The Theory of Matrices, Second Edition*, Academic Press, New York.

C. Lanczos (1950). "An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators," *J. Res. Nat. Bur. Stand. 45*, 255–82.

B. Lang (1996). "Parallel Reduction of Banded Matrices to Bidiagonal Form," *Parallel Computing 22*, 1–18.

J. Larson and A. Sameh (1978). "Efficient Calculation of the Effects of Roundoff Errors," *ACM Trans. Math. Soft. 4*, 228–36.

A. Laub (1981). "Efficient Multivariable Frequency Response Computations," *IEEE Trans. Auto. Cont. AC-26*, 407–8.

A. Laub(1985). "Numerical Linear Algebra Aspects of Control Design Computations," *IEEE Trans. Auto. Cont. AC-30*, 97–108.

C.L. Lawson and R.J. Hanson (1969). "Extensions and Applications of the Householder Algorithm for Solving Linear Least Squares Problems," *Math. Comp. 23*, 787–812.

C.L. Lawson and R.J. Hanson (1974). *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ. Reprinted with a detailed "new developments" appendix in 1996 by SIAM Publications, Philadelphia, PA.

C.L. Lawson, R.J. Hanson, , D.R. Kincaid, and F.T. Krogh (1979). "Basic Linear Algebra Subprograms for FORTRAN Usage," *ACM Trans. Math. Soft. 5*, 308–323.

C.L. Lawson, R.J. Hanson, D.R. Kincaid, and F.T. Krogh (1979). "Algorithm 539, Basic Linear Algebra Subprograms for FORTRAN Usage," *ACM Trans. Math. Soft. 5*, 324–325.

D. Lay (1994). *Linear Algebra and Its Applications*, Addison-Wesley, Reading, MA.

N.J. Lehmann (1963). "Optimale Eigenwerteinschliessungen," *Numer. Math. 5*, 246–72.

R.B, Lehoucq (1995). "Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration," Ph.D. thesis, Rice University, Houston Texas.

R.B. Lehoucq (1996). "Restarting an Arnoldi Reduction," Report MCS-P591-0496, Argonne National Laboratory, Argonne Illinois.

R.B. Lehoucq and D.C. Sorensen (1996). "Deflation Techniques for an Implicitly Restarted Iteration," *SIAM J. Matrix Analysis and Applic*, to appear.

F.T. Leighton (1992). *Introduction to Parallel Algorithms and Architectures*, Morgan Kaufmann, San Mateo, CA.

F. Lemeire (1973). "Bounds for Condition Numbers of Triangular Value of a Matrix," *Lin. Alg. and Its Applic. 11*, 1–2.

S.J. Leon (1980). *Linear Algebra with Applications*. Macmillan, New York.

S.J. Leon (1994). "Maximizing Bilinear Forms Subject to Linear Constraints," *Lin. Alg. and Its Applic. 210*, 49–58.

N. Levinson (1947). "The Weiner RMS Error Criterion in Filter Design and Prediction," *J. Math. Phys. 25*, 261–78.

J. Lewis, ed. (1994). *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, SIAM Publications, Philadelphia, PA.

G. Li and T. Coleman (1988). "A Parallel Triangular Solver for a Distributed-Memory Multiprocessor," *SIAM J. Sci. and Stat. Comp. 9*, 485–502.

K. Li and T-Y. Li (1993). "A Homotopy Algorithm for a Symmetric Generalized Eigenproblem," *Numerical Algorithms 4*, 167–195.

K. Li, T-Y. Li, and Z. Zeng (1994). "An Algorithm for the Generalized Symmetric Tridiagonal Eigenvalue Problem," *Numerical Algorithms 8*, 269–291.

R-C. Li (1993). "Bounds on Perturbations of Generalized Singular Values and of Associated Subspaces," *SIAM J. Matrix Anal. Appl. 14*, 195–234.

R-C. Li (1994). "On Eigenvalue Variations of Rayleigh Quotient Matrix Pencils of a Definite Pencil," *Lin. Alg. and Its Applic. 208/209*, 471–483.

R-C. Li (1995). "New Perturbation Bounds for the Unitary Polar Factor," *SIAM J. Matrix Anal. Appl. 16*, 327–332.

R.-C. Li (1996). "Relative Perturbation Theory (I) Eigenvalue and Singular Value Variations," Technical Report UCB//CSD-94-855, Department of EECS, University of California at Berkeley.

R.-C. Li (1996). "Relative Perturbation Theory (II) Eigenspace and Singular Subspace Variations," Technical Report UCB//CSD-94-856, Department of EECS, University of California at Berkeley.

Y. Li (1993). "A Globally Convergent Method for L_p Problems," *SIAM J. Optimization 3*, 609–629.

W-W. Lin and C.W. Chen (1991). "An Acceleration Method for Computing the Generalized Eigenvalue Problem on a Parallel Computer," *Lin.Alg. and Its Applic. 146*, 49–65.

I. Linnik (1961). *Method of Least Squares and Principles of the Theory of Observations*, Pergamon Press, New York.

E. Linzer (1992). "On the Stability of Solution Methods for Band Toeplitz Systems," *Lin.Alg. and Its Applic. 170*, 1–32.

S. Lo, B. Philippe, and A. Sameh (1987). "A Multiprocessor Algorithm for the Symmetric Tridiagonal Eigenvalue Problem," *SIAM J. Sci. and Stat. Comp. 8*, s155-s165.

G. Loizou (1969). "Nonnormality and Jordan Condition Numbers of Matrices," *J. ACM 16*, 580–40.

G. Loizou (1972). "On the Quadratic Convergence of the Jacobi Method for Normal Matrices," *Comp. J. 15*, 274–76.

M. Lotkin (1956). "Characteristic Values of Arbitrary Matrices," *Quart. Appl. Math. 14*, 267–75.

H. Lu (1994). "Fast Solution of Confluent Vandermonde Linear Systems," *SIAM J. Matrix Anal. Appl. 15*, 1277–1289.

H. Lu (1996). "Solution of Vandermonde-like Systems and Confluent Vandermonde-like Systems," *SIAM J. Matrix Anal. Appl. 17*, 127–138.

D. G. Luenberger (1973). *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, New York.

F.T. Luk (1978). "Sparse and Parallel Matrix Computations," PhD Thesis, Report STANS-CS-78-685, Department of Computer Science, Stanford University, Stanford, CA.

F.T. Luk (1980). "Computing the Singular Value Decomposition on the ILLIAC IV," *ACM Trans. Math. Soft. 6*, 524–39.

F.T. Luk (1986). "A Rotation Method for Computing the QR Factorization," *SIAM J. Sci. and Stat. Comp. 7*, 452–459.

F.T. Luk (1986). "A Triangular Processor Array for Computing Singular Values," *Lin. Alg. and Its Applic. 77*, 259–274.

N. Mackey (1995). "Hamilton and Jacobi Meet Again: Quaternions and the Eigenvalue Problem," *SIAM J. Matrix Anal. Appl. 16*, 421–435.

A. Madansky (1959). "The Fitting of Straight Lines When Both Variables Are Subject to Error," *J. Amer. Stat. Assoc. 54*, 173–205.

N. Madsen, G. Roderigue, and J. Karush (1976). "Matrix Multiplication by Diagonals on a Vector Parallel Processor," *Infomation Processing Letters 5*, 41-45.

K.N. Majinder (1979). "Linear Combinations of Hermitian and Real Symmetric Matrices," *Lin. Alg. and Its Applic. 25*, 95–105.

J. Makhoul (1975). "Linear Prediction: A Tutorial Review," *Proc. IEEE* 63(4), 561-80.

M.A. Malcolm and J. Palmer (1974). "A Fast Method for Solving a Class of Tridiagonal Systems of Linear Equations," *Comm. ACM 17*, 14–17.

L. Mansfield (1991). "Damped Jacobi Preconditioning and Coarse Grid Deflation for Conjugate Gradient Iteration on Parallel Computers," *SIAM J. Sci. and Stat. Comp. 12*, 1314–1323.

T.A. Manteuffel (1977). "The Tchebychev Iteration for Nonsymmetric Linear Systems," *Numer. Math. 28*, 307–27.

T.A. Mantueffel (1979). " Shifted Incomplete Cholesky Factorization," in *Sparse Matrix Proceedings* , 1978, ed. I.S. Duff and G.W. Stewart, SIAM Publications, Philadelphia, PA.

M. Marcus (1993). *Matrices and* MATLAB*: A Tutorial*, Prentice Hall, Upper Saddle River, NJ.

M. Marcus and H. Minc (1964). *A Survey of Matrix Theory and Matrix Inequalities*, Allyn and Bacon, Boston.

J. Markel and A. Gray (1976). *Linear Prediction of Speech*, Springer-Verlag, Berlin and New York.

M. Marrakchi and Y. Robert (1989). "Optimal Algorithms for Gaussian Elimination on an MIMD Computer," *Parallel Computing 12*, 183–194.

R.S. Martin, G. Peters, and J.H. Wilkinson (1970). "The QR Algorithm for Real Hessenberg Matrices," *Numer. Math. 14*, 219–31. See also Wilkinson and Reinsch(1971, pp. 359–71).

R.S. Martin and J.H. Wilkinson (1965). "Symmetric Decomposition of Positive Definite Band Matrices," *Numer. Math. 7*, 355–61.

R.S. Martin and J.H. Wilkinson (1967). "Solution of Symmetric and Unsymmetric Band Equations and the Calculation of Eigenvectors of Band Matrices," *Numer. Math. 9*, 279–301. See also Wilkinson and Reinsch (1971, pp.70–92).

R.S. Martin and J.H. Wilkinson (1968). "Similarity Reduction of a General Matrix to Hessenberg Form," *Numer. Math.* *12*, 349–68. See also Wilkinson and Reinsch (1971,pp.339–58).

R.S. Martin and J.H. Wilkinson (1968). "The Modified LR Algorithm for Complex Hessenberg Matrices," *Numer. Math.* *12*, 369–76. See also Wilkinson and Reinsch(1971, pp. 396–403).

R.S. Martin and J.H. Wilkinson (1968). "Householder's Tridiagonalization of a Symmetric Matrix," *Numer. Math.* *11*, 181-95. See also Wilkinson and Reinsch (1971, pp.212–26).

R.S. Martin and J.H. Wilkinson (1968). "Reduction of a Symmetric Eigenproblem $Ax = \lambda Bx$ and Related Problems to Standard Form," *Numer. Math.* *11*, 99–110.

R.S. Martin, G. Peters, and J.H. Wilkinson (1965). "Symmetric Decomposition of a Positive Definite Matrix," *Numer. Math.* *7*, 362-83.

R.S. Martin, G. Peters, and J.H. Wilkinson (1966). "Iterative Refinement of the Solution of a Positive Definite System of Equations," *Numer. Math.* *8*, 203–16.

R.S. Martin, C. Reinsch, and J.H. Wilkinson (1970). "The QR Algorithm for Band Symmetric Matrices," *Numer. Math.* *16*, 85–92. See also See also Wilkinson and Reinsch (1971, pp.266–72).

W.F. Mascarenhas (1994). "A Note on Jacobi Being More Accurate than QR," *SIAM J. Matrix Anal. Appl.* *15*, 215–218.

R. Mathias (1992). "Matrices with Positive Definite Hermitian Part: Inequalities and Linear Systems," *SIAM J. Matrix Anal. Appl.* *13*, 640–654.

R. Mathias (1992). "Evaluating the Frechet Derivative of the Matrix Exponential," *Numer. Math.* *63*, 213–226.

R. Mathias (1993). "Approximation of Matrix-Valued Functions," *SIAM J. Matrix Anal. Appl.* *14*, 1061–1063.

R. Mathias (1993). "Perturbation Bounds for the Polar Decomposition," *SIAM J. Matrix Anal. Appl.* *14*, 588–597.

R. Mathias (1995). "Accurate Eigensystem Computations by Jacobi Methods," *SIAM J. Matrix Anal. Appl.* *16*, 977–1003.

R. Mathias (1995). "The Instability of Parallel Prefix Matrix Multiplication," *SIAM J. Sci. Comp.* *16*, 956–973.

R. Mathias and G.W. Stewart (1993). "A Block QR Algorithm and the Singular Value Decomposition," *Lin. Alg. and Its Applic.* *182*, 91–100.

K. Mathur and S.L. Johnsson (1994). "Multiplication of Matrices of Arbitrary Shape on a Data Parallel Computer," *Parallel Computing* *20*, 919–952.

B. Mattingly, C. Meyer, and J. Ortega (1989). "Orthogonal Reduction on Vector Computers," *SIAM J. Sci. and Stat. Comp.* *10*, 372–381.

O. McBryan and E.F. van de Velde (1987). "Hypercube Algorithms and Implementations," *SIAM J. Sci. and Stat. Comp.* *8*, s227–s287.

C. McCarthy and G. Strang (1973). "Optimal Conditioning of Matrices," *SIAM J. Num. Anal.* *10*, 370–88.

S.F. McCormick (1972). "A General Approach to One-Step Iterative Methods with Application to Eigenvalue Problems," *J. Comput. Sys. Sci.* *6*, 354–72.

W.M. McKeeman (1962). "Crout with Equilibration and Iteration," *Comm. ACM.* *5*, 553–55.

K. Meerbergen, A. Spence, and D. Roose (1994). "Shift-Invert and Cayley Transforms for the Detection of Rightmost Eigenvalues of Nonsymmetric Matrices," *BIT* *34*, 409–423.

V. Mehrmann (1988). "A Symplectic Orthogonal Method for Single Input or Single Output Discrete Time Optimal Quadratic Control Problems," *SIAM J. Matrix Anal. Appl.* *9*, 221–247.

V. Mehrmann (1993). "Divide and Conquer Methods for Block Tridiagonal Systems," *Parallel Computing* *19*, 257–280.

U. Meier (1985). "A Parallel Partition Method for Solving Banded Systems of Linear Equations," *Parallel Computers* *2*, 33–43.

U. Meier and A. Sameh (1988). "The Behavior of Conjugate Gradient Agorithms on a Multivector Processor with a Hierarchical Memory," *J. Comput. Appl. Math. 24*, 13–32.

J.A. Meijerink and H.A. Van der vorst (1977). "An Iterative Solution Method for Linear Equation Systems of Which the Coefficient Matrix is a Symmetric M-Matrix," *Math. Comp. 31*, 148–62.

J. Meinguet (1983). "Refined Error Analyses of Cholesky Factorization," *SIAM J. Numer. Anal. 20*, 1243–1250.

R. Melhem(1987). "Toward Efficient Implementation of Preconditioned Conjugate Gradient Methods on Vector Supercomputers," *Int'l J. Supercomputing Applications 1*, 70–98.

M.L. Merriam (1985). "On the Factorization of Block Tridiagonals With Storage Constraints," *SIAM J. Sci. and Stat. Comp. 6*, 182-192.

G. Meurant (1984). "The Block Preconditioned Conjugate Gradient Method on Vector Computers," *BIT 24*, 623–633.

G. Meurant (1989). "Domain Decomposition Methods for Partial Differential Equations on Parallel Computers," to appear *Int'l J. Supercomputing Applications.*

G. Meurant (1992). "A Review on the Inverse of Symmetric Tridiagonal and Block Tridiagonal Matrices," *SIAM J. Matrix Anal. Appl. 13*, 707–728.

C.D. Meyer (1997). *A Course in Applied Linear Algebra*, to be published.

C.D. Meyer and G.W. Stewart (1988). "Derivatives and Perturbations of Eigenvectors," *SIAM J. Num. Anal. 25*, 679–691.

W. Miller (1975). "Computational Complexity and Numerical Stability," *SIAM J. Computing 4*, 97–107.

W. Miller and D. Spooner (1978). "Software for Roundoff Analysis, II," *ACM Trans. Math. Soft. 4*, 369–90.

G. Miminis and C.C. Paige (1982). "An Algorithm for Pole Assignment of Time Invariant Linear Systems," *International J. of Control 35*, 341–354.

L. Mirsky (1960). "Symmetric Gauge Functions and Unitarily Invariant Norms," *Quart. J. Math. 11*, 50-59.

L. Mirsky (1963). *An Introduction to Linear Algebra*, Oxford University Press, Oxford.

J.J. Modi (1988).*Parallel Algorithms and Matrix Computation*, Oxford University Press, Oxford.

J.J. Modi and M.R.B. Clarke (1986). "An Alternative Givens Ordering," *Numer. Math. 43*, 83–90.

J.J. Modi and J.D. Pryce (1985). "Efficient Implementation of Jacobi's Diagonalization Method on the DAP," *Numer. Math. 46*, 443–454.

C.B. Moler (1967). "Iterative Refinement in Floating Point," *J. ACM 14*, 316-71.

C.B. Moler and D. Morrison (1983). "Singular Value Analysis of Cryptograms," *Amer. Math. Monthly 90*, 78–87.

C.B. Moler and G.W. Stewart (1973). "An Algorithm for Generalized Matrix Eigenvalue Problems," *SIAM J. Num. Anal. 10*, 241–56.

C.B. Moler and C.F. Van Loan (1978). "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review 20*, 801–36.

R. Montoye and D. Laurie (1982). "A Practical Algorithm for the Solution of Triangular Systems on a Parallel Processing System," *IEEE Trans. Comp. C-31*, 1076–1082.

M.S. Moonen and B. De Moor, eds. (1995). *SVD and Signal Processing III: Algorithms, Analysis, and Applications.* Elsevier, Amsterdam.

M.S. Moonen, G.H. Golub, and B.L.R. de Moor, eds. (1993). *Linear Algebra for Large Scale and Real-Time Applications*, Kluwer, Dordrecht, The Netherlands.

M.S. Moonen, P. Van Dooren, and J. Vandewalle (1992). "A Singular Value Decomposition Updating Algorithm," *SIAM J. Matrix Anal. Appl. 13*, 1015–1038.

R.B. Morgan (1995). "A Restarted GMRES Method Augmented with Eigenvectors," *SIAM J. Matrix Anal. Applic. 16*, 1154–1171.

R.B. Morgan (1996). "On Restarting the Arnoldi Method for Large Scale Eigenvalue Problems," *Math Comp*, to appear.

M. Mu. (1995). "A New family of Preconditioners for Domain Decomposition," *SIAM J. Sci. Comp.* *16*, 289–306.
D. Mueller (1966). "Householder's Method for Complex Matrices and Hermitian Matrices," *Numer. Math.* *8*, 72–92.
F.D. Murnaghan and A. Wintner (1931). "A Canonical Form for Real Matrices Under Orthogonal Transformations," *Proc. Nat. Acad. Sci.* *17*, 417–20.
N. Nachtigal, S. Reddy, and L. Trefethen (1992). "How Fast Are Nonsymmetric Matrix Iterations," *SIAM J. Matrix Anal. Appl.* *13*, 778–795.
N. Nachtigal, L. Reichel, and L. Trefethen (1992). "A Hybrid GMRES Algorithm for Nonsymmetric Linear Systems," *SIAM J. Matrix Anal. Appl.* *13*, 796–825.
T. Nanda (1985). "Differential Equations and the QR Algorithm," *SIAM J. Numer. Anal.* *22*, 310–321.
J.C. Nash (1975). "A One-Sided Tranformation Method for the Singular Value Decomposition and Algebraic Eigenproblem," *Comp. J.* *18*, 74–76.
M.Z. Nashed (1976). *Generalized Inverses and Applications*, Academic Press, New York.
R.A. Nicolaides (1974). "On a Geometrical Aspect of SOR and the Theory of Consistent Ordering for Positive Definite Matrices," *Numer. Math.* *12*, 99–104.
W. Niethammer and R.S. Varga (1983). "The Analysis of k-step Iterative Methods for Linear Systems from Summability Theory," *Numer. Math.* *41*, 177–206.
B. Noble and J.W. Daniel (1977). *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs.
Y. Notay (1992). "On the Robustness of Modified Incomplete Factorization Methods," *J. Comput. Math.* *40*, 121–141.
C. Oara (1994). "Proper Deflating Subspaces: Properties, Algorithms, and Applications," *Numerical Algorithms* *7*, 355–373.
W. Oettli and W. Prager (1964). "Compatibility of Approximate Solutions of Linear Equations with Given Error Bounds for Coefficients and Right Hand Sides," *Numer. Math.* *6*, 405–409.
D.P. O'Leary (1980). "Estimating Matrix Condition Numbers,"*SIAM J. Sci. Stat. Comp.* *1*, 205–9.
D.P. O'Leary (1980). "The Block Conjugate Gradient Algorithm and Related Methods," *Lin. Alg. and Its Applic.* *29*, 293–322.
D.P. O'Leary (1987). "Parallel Implementation of the Block Conjugate Gradient Algorithm," *Parallel Computers* *5*, 127–140.
D.P. O'Leary (1990). "On Bounds for Scaled Projections and Pseudoinverses," *Lin. Alg. and Its Applic.* *132*, 115–117.
D.P. O'Leary and J.A. Simmons (1981). "A Bidiagonalization-Regularization Procedure for Large Scale Discretizations of Ill-Posed Problems," *SIAM J. Sci. and Stat. Comp.* *2*, 474–489.
D.P. O'Leary and G.W. Stewart (1985). "Data Flow Algorithms for Parallel Matrix Computations," *Comm. ACM* *28*, 841-853.
D.P. O'Leary and G.W. Stewart (1986). "Assignment and Scheduling in Parallel Matrix Factorization," *Lin. Alg. and Its Applic.* *77*, 275–300.
S.J. Olszanskyj, J.M. Lebak, and A.W. Bojanczyk (1994). "Rank-k Modification Methods for Recursive Least Squares Problems," *Numerical Algorithms* *7*, 325–354.
A.V. Oppenheim (1978). *Applications of Digital Signal Processing* , Prentice-Hall, Englewood Cliffs.
J.M. Ortega (1987). *Matrix Theory: A Second Course*, Plenum Press, New York.
J.M. Ortega (1988). "The ijk Forms of Factorization Methods I: Vector Computers," *Parallel Computers* *7*, 135–147.
J.M. Ortega (1988). *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York.
J.M. Ortega and C.H. Romine (1988). "The ijk Forms of Factorization Methods II: Parallel Systems," *Parallel Computing* *7*, 149–162.
J.M. Ortega and R.G. Voigt (1985). "Solution of Partial Differential Equations on Vector and Parallel Computers," *SIAM Review* *27*, 149–240.

E.E. Osborne (1960). "On Preconditioning of Matrices," *JACM 7*, 338–45.
M.H.C. Paardekooper (1971). "An Eigenvalue Algorithm for Skew Symmetric Matrices," *Numer. Math. 17*, 189–202.
M.H.C. Paardekooper (1991). "A Quadratically Convergent Parallel Jacobi Process for Diagonally Dominant Matrices with Nondistinct Eigenvalues," *Lin.Alg. and Its Applic. 145*, 71–88.
C.C. Paige (1970). "Practical Use of the Symmetric Lanczos Process with Reorthogonalization," *BIT 10*, 183–95.
C.C. Paige (1971). "The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices," Ph.D. thesis, London University.
C.C. Paige (1972). "Computational Variants of the Lanczos Method for the Eigenproblem," *J. Inst. Math. Applic. 10*, 373–81.
C.C. Paige (1973). "An Error Analysis of a Method for Solving Matrix Equations," *Math. Comp. 27*, 355–59.
C.C. Paige (1974). "Bidiagonalization of Matrices and Solution of Linear Equations," *SIAM J. Num. Anal. 11*, 197–209.
C.C. Paige (1974). "Eigenvalues of Perturbed Hermitian Matrices," *Lin. Alg. and Its Applic* . 8, 1–10.
C.C. Paige (1976). "Error Analysis of the Lanczos Algorithm for Tridiagonalizing Symmetric Matrix," *J. Inst. Math. Applic. 18*, 341–49.
C.C. Paige (1979). "Computer Solution and Perturbation Analysis of Generalized Least Squares Problems," *Math. Comp. 33*, 171–84.
C.C. Paige (1979). "Fast Numerically Stable Computations for Generalized Linear Least Squares Problems," *SIAM J. Num. Anal. 16*, 165–71.
C.C. Paige (1980). "Accuracy and Effectiveness of the Lanczos Algorithm for the Symmetric Eigenproblem," *Lin. Alg. and Its Applic. 34*, 235–58.
C.C. Paige (1981). "Properties of Numerical Algorithms Related to Computing Controllability," *IEEE Trans. Auto. Cont. AC-26*, 130–38.
C.C. Paige (1984). "A Note on a Result of Sun J.-Guang: Sensitivity of the CS and GSV Decompositions," *SIAM J. Numer. Anal. 21*, 186–191.
C.C. Paige (1985). "The General Linear Model and the Generalized Singular Value Decomposition," *Lin. Alg. and Its Applic. 70*, 269–284.
C.C. Paige (1986). "Computing the Generalized Singular Value Decomposition," *SIAM J. Sci. and Stat. Comp. 7*, 1126–1146.
C.C. Paige (1990). "Some Aspects of Generalized QR Factorization," in *Reliable Numerical Computations*, M. Cox and S. Hammarling (eds), Clarendon Press, Oxford.
C.C. Paige, B.N. Parlett,and H.A. Van Der Vorst (1995). "Approximate Solutions and Eigenvalue Bounds from Krylov Subspaces," *Numer. Linear Algebra with Applic. 2*, 115–134.
C.C. Paige and M.A. Saunders (1975). "Solution of Sparse Indefinite Systems of Linear Equations," *SIAM J. Num. Anal. 12*, 617–29.
C.C. Paige and M. Saunders (1981). "Toward a Generalized Singular Value Decomposition," *SIAM J. Num. Anal. 18*, 398–405.
C.C. Paige and M.A. Saunders (1982). "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares," *ACM Trans. Math. Soft. 8*, 43–71.
C.C. Paige and M.A. Saunders (1982). "Algorithm 583 LSQR: Sparse Linear Equations and Least Squares Problems," *ACM Trans. Math. Soft. 8*, 195–209.
C.C. Paige and P. Van Dooren (1986). "On the Quadratic Convergence of Kogbetliantz's Algorithm for Computing the Singular Value Decomposition," *Lin. Alg. and Its Applic. 77*, 301–313.
C.C. Paige and C. Van Loan (1981). "A Schur Decomposition for Hamiltonian Matrices," *Lin. Alg. and Its Applic. 41*, 11–32.
C.C. Paige and M. Wei (1993). "Analysis of the Generalized Total Least Squares Problem $AX = B$ when Some of the Columns are Free of Error," *Numer. Math. 65*, 177–202.
C.C. Paige and M. Wei (1994). "History and Generality of the CS Decomposition," *Lin. Alg. and Its Applic. 208/209*, 303–326.

C.-T. Pan (1993). "A Perturbation Analysis of the Problem of Downdating a Cholesky Factorization," *Lin. Alg. and Its Applic. 183*, 103–115.

V. Pan (1984). "How Can We Speed Up Matrix Multiplication?," *SIAM Review 26*, 393–416.

H. Park (1991). "A Parallel Algorithm for the Unbalanced Orthogonal Procrustes Problem," *Parallel Computing 17*, 913–923.

H. Park and L. Eldén (1995). "Downdating the Rank-Revealing URV Decomposition," *SIAM J. Matrix Anal. Appl. 16*, 138–155.

B.N. Parlett (1965). "Convergence of the Q-R Algorithm," *Numer. Math. 7*, 187–93. (Correction in *Numer. Math. 10*, 163–64.)

B.N. Parlett (1966). "Singular and Invariant Matrices Under the QR Algorithm," *Math. Comp. 20*, 611–15.

B.N. Parlett (1967). "Canonical Decomposition of Hessenberg Matrices," *Math. Comp. 21*, 223–27.

B.N. Parlett (1968). "Global Convergence of the Basic QR Algorithm on Hessenberg Matrices," *Math. Comp. 22*, 803–17.

B.N. Parlett (1971). "Analysis of Algorithms for Reflections in Bisectors," *SIAM Review 13*, 197–208.

B.N. Parlett (1974). "The Rayleigh Quotient Iteration and Some Generalizations for Nonnormal Matrices," *Math. Comp. 28*, 679-93.

B.N. Parlett (1976). "A Recurrence Among the Elements of Functions of Triangular Matrices," *Lin. Alg. and Its Applic. 14*, 117–21.

B.N. Parlett (1980). *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ.

B.N. Parlett (1980). "A New Look at the Lanczos Algorithm for Solving Symmetric Systems of Linear Equations," *Lin. Alg. and Its Applic. 29*, 323–46.

B.N. Parlett (1992). "Reduction to Tridiagonal Form and Minimal Realizations," *SIAM J. Matrix Anal. Appl. 13*, 567–593.

B.N. Parlett (1995). "The New qd Algorithms," *ACTA Numerica 5*, 459–491.

B.N. Parlett and B. Nour-Omid (1985). "The Use of a Refined Error Bound When Updating Eigenvalues of Tridiagonals," *Lin. Alg. and Its Applic. 68*, 179–220.

B.N. Parlett and W.G. Poole (1973). "A Geometric Theory for the QR, LU, and Power Iterations," *SIAM J. Num. Anal. 10*, 389–412.

B.N. Parlett and J.K. Reid (1970). "On the Solution of a System of Linear Equations Whose Matrix is Symmetric but not Definite," *BIT 10*, 386–97.

B.N. Parlett and J.K. Reid (1981). "Tracking the Progress of the Lanczos Algorithm for Large Symmetric Eigenproblems," *IMA J. Num. Anal. 1*, 135–55.

B.N. Parlett and C. Reinsch (1969). "Balancing a Matrix for Calculation of Eigenvalues and Eigenvectors," *Numer. Math. 13*, 292–304. See also Wilkinson and Reinsch(1971, pp. 315-26).

B.N. Parlett and R. Schreiber (1988). "Block Reflectors: Theory and Computation," *SIAM J. Num. Anal. 25*, 189–205.

B.N. Parlett and D.S. Scott (1979). "The Lanczos Algorithm with Selective Orthogonalization," *Math. Comp. 33*, 217–38.

B.N. Parlett, H. Simon, and L.M. Stringer (1982). "On Estimating the Largest Eigenvalue with the Lanczos Algorithm," *Math. Comp. 38*, 153–166.

B.N. Parlett, D. Taylor, and Z. Liu (1985). "A Look-Ahead Lanczos Algorithm for Unsymmetric Matrices," *Math. Comp. 44*, 105–124.

N. Patel and H. Jordan (1984). "A Parallelized Point Rowwise Successive Over-Relaxation Method on a Multiprocessor," *Parallel Computing 1*, 207–222.

R.V. Patel, A.J. Laub, and P.M. Van Dooren, eds. (1994). *Numerical Linear Algebra Techniques for Systems and Control*, IEEE Press, Piscataway, New Jersey.

D.A. Patterson and J.L. Hennessy (1989). *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, Inc., Palo Alto, CA.

M.S. Paterson and L.J. Stockmeyer (1973). "On the Number of Nonscalar Multiplications Necessary to Evaluate Polynomials," *SIAM J. Comp. 2*, 60–66.

K. Pearson (1901). "On Lines and Planes of Closest Fit to Points in Space," *Phil. Mag. 2*, 559–72.

G. Peters and J.H. Wilkinson (1969). "Eigenvalues of $Ax = \lambda Bx$ with Band Symmetric A and B," *Comp. J. 12*, 398-404.

G. Peters and J.H. Wilkinson (1970). "The Least Squares Problem and Pseudo-Inverses," *Comp. J. 13*, 309–16.

G. Peters and J.H. Wilkinson (1970). "$Ax = \lambda Bx$ and the Generalized Eigenproblem," *SIAM J. Num. Anal. 7*, 479–92.

G. Peters and J.H. Wilkinson (1971). "The Calculation of Specified Eigenvectors by Inverse Iteration," in Wilikinson and Reinsch (1971, pp.418–39).

G. Peters and J.H. Wilkinson (1979). "Inverse Iteration, Ill-Conditioned Equations, and Newton's Method," *SIAM Review 21*, 339–60.

D.J. Pierce and R.J. Plemmons (1992). "Fast Adaptive Condition Estimation," *SIAM J. Matrix Anal. Appl. 13*, 274–291.

S. Pissanetsky (1984). *Sparse Matrix Technology*, Academic Press, New York.

R.J. Plemmons (1974). "Linear Least Squares by Elimination and MGS," *J. Assoc. Comp. Mach. 21*, 581–85.

R.J. Plemmons (1986). "A Parallel Block Iterative Scheme Applied to Computations in Structural Analysis," *SIAM J. Alg. and Disc. Methods 7*, 337–347.

R.J. Plemmons and C.D. Meyer, eds. (1993). *Linear Algebra, Markov Chains, and Queuing Models*, Springer-Verlag, New York.

A. Pokrzywa (1986). "On Perturbations and the Equivalence Orbit of a Matrix Pencil," *Lin. Alg. and Applic. 82*, 99–121.

E.L. Poole and J.M. Ortega (1987). "Multicolor ICCG Methods for Vector Computers," *SIAM J. Numer. Anal. 24*, 1394–1418.

D.A. Pope and C. Tompkins (1957). "Maximizing Functions of Rotations: Experiments Concerning Speed of Diagonalization of Symmetric Matrices Using Jacobi's Method," *J. ACM 4*, 459–66.

A. Pothen, S. Jha, and U. Vemapulati (1987). "Orthogonal Factorization on a Distributed Memory Multiprocessor," in *Hypercube Multiprocessors*, ed. M.T. Heath, SIAM Publications, 1987.

M.J.D. Powell and J.K. Reid (1968). "On Applying Householder's Method to Linear Least Squares Problems," *Proc. IFIP Congress*, pp. 122–26.

R. Pratap (1995). *Getting Started with* MATLAB, Saunders College Publishing, Fort Worth, TX.

J.D. Pryce (1984). "A New Measure of Relative Error for Vectors," *SIAM J. Num. Anal. 21*, 202-21.

C. Puglisi (1992). "Modification of the Householder Method Based on the Compact WY Representation," *SIAM J. Sci. and Stat. Comp. 13*, 723–726.

S. Qiao(1986). "Hybrid Algorithm for Fast Toeplitz Orthogonalization," *Numer. Math. 53*, 351–366.

S. Qiao (1988). "Recursive Least Squares Algorithm for Linear Prediction Problems," *SIAM J. Matrix Anal. Appl. 9*, 323–328.

C.M. Rader and A.O. Steinhardt (1988). "Hyperbolic Householder Transforms," *SIAM J. Matrix Anal. Appl. 9*, 269–290.

G. Radicati di Brozolo and Y. Robert (1989). "Parallel Conjugate Gradient-like Algorithms for Solving Sparse Nonsymmetric Linear Systems on a Vector Multiprocessor," *Parallel Computing 11*, 233-240.

P. Raghavan (1995). "Distributed Sparse Gaussian Elimination and Orthogonal Factorization," *SIAM J. Sci. Comp. 16*, 1462–1477.

W. Rath (1982). "Fast Givens Rotations for Orthogonal Similarity," *Numer. Math. 40*, 47–56.

P.A. Regalia and S. Mitra (1989). "Kronecker Products, Unitary Matrices, and Signal Processing Applications," *SIAM Review 31*, 586–613.

L. Reichel (1991). "Fast QR Decomposition of Vandermonde-Like Matrices and Polynomial Least Squares Approximation," *SIAM J. Matrix Anal. Appl. 12*, 552–564.

L. Reichel and L.N. Trefethen (1992). "Eigenvalues and Pseudo-eigenvalues of Toeplitz Matrices," *Lin. Alg. and Its Applic. 162/163/164*, 153–186.

J.K. Reid (1967). "A Note on the Least Squares Solution of a Band System of Linear Equations by Householder Reductions," *Comp. J. 10,* 188–89.

J.K. Reid (1971). "A Note on the Stability of Gaussian Elimination," *J. Inst. Math. Applic. 8,* 374–75.

J.K. Reid (1971). " On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations," *in Large Sparse Sets of Linear Equations* , ed. J.K. Reid, Academic Press, New York, pp. 231–54.

J.K. Reid (1972). "The Use of Conjugate Gradients for Systems of Linear Equations Possessing Property A," *SIAM J. Num. Anal. 9,* 325–32.

C. Reinsch and F.L. Bauer (1968). "Rational QR Transformation with Newton's Shift for Symmetric Tridiagonal Matrices," *Numer. Math. 11,* 264–72. See also Wilkinson and Reinsch (1971, pp.257–65).

J.R. Rice (1966). "A Theory of Condition," *SIAM J. Num. Anal. 3,* 287–310.

J.R. Rice (1966). "Experiments on Gram-Schmidt Orthogonalization," *Math. Comp. 20,* 325–28.

J.R. Rice (1981). *Matrix Computations and Mathematical Software*, Academic Press, New York.

R.F. Rinehart (1955). "The Equivalence of Definitions of a Matric Function," *Amer. Math. Monthly 62,* 395–414.

Y. Robert (1990). *The Impact of Vector and Parallel Architectures on the Gaussian Elimination Algorithm*, Halsted Press, New York.

H.H. Robertson (1977). "The Accuracy of Error Estimates for Systems of Linear Algebraic Equations," *J. Inst. Math. Applic. 20,* 409–14.

G. Rodrigue (1973). "A Gradient Method for the Matrix Eigenvalue Problem $Ax = \lambda Bx$," *Numer. Math. 22,* 1–16.

G. Rodrigue, ed. (1982). *Parallel Computation*, Academic Press, New York.

G. Rodrigue and D. Wolitzer (1984). "Preconditioning by Incomplete Block Cyclic Reduction," *Math. Comp. 42,* 549–566.

C.H. Romine and J.M. Ortega (1988). "Parallel Solution of Triangular Systems of Equations," *Parallel Computing 6,* 109–114.

D.J. Rose (1969). "An Algorithm for Solving a Special Class of Tridiagonal Systems of Linear Equations," *Comm. ACM 12,* 234–36.

D.J. Rose and R. A. Willoughby, eds. (1972). *Sparse Matrices and Their Applications*, Plenum Press, New York, 1972

A. Ruhe (1968). On the Quadratic Convergence of a Generalization of the Jacobi Method to Arbitrary Matrices," *BIT 8,* 210–31.

A. Ruhe (1969). "The Norm of a Matrix After a Similarity Transformation," *BIT 9,* 53–58.

A. Ruhe (1970). "An Algorithm for Numerical Determination of the Structure of a General Matrix," *BIT 10,* 196–216.

A. Ruhe (1970). "Perturbation Bounds for Means of Eigenvalues and Invariant Subspaces," *BIT 10,* 343–54.

A. Ruhe (1970). "Properties of a Matrix with a Very Ill-Conditioned Eigenproblem," *Numer. Math. 15,* 57–60.

A. Ruhe (1972). "On the Quadratic Convergence of the Jacobi Method for Normal Matrices," *BIT 7,* 305–13.

A. Ruhe (1974). "SOR Methods for the Eigenvalue Problem with Large Sparse Matrices," *Math. Comp.* 28, 695–710.

A. Ruhe (1975). "On the Closeness of Eigenvalues and Singular Values for Almost Normal Matrices," *Lin. Alg. and Its Applic. 11,* 87–94.

A. Ruhe (1979). "Implementation Aspects of Band Lanczos Algorithms for Computation of Eigenvalues of Large Sparse Symmetric Matrices," *Math. Comp. 33,* 680-87.

A. Ruhe (1983). "Numerical Aspects of Gram-Schmidt Orthogonalization of Vectors," *Lin. Alg. and Its Applic. 52/53,* 591–601.

A. Ruhe (1984). "Rational Krylov Algorithms for Eigenvalue Computation," *Lin. Alg. and Its Applic. 58*, 391–405.

A. Ruhe (1987). "Closest Normal Matrix Found!," *BIT 27*, 585-598.

A. Ruhe (1994). "Rational Krylov Algorithms for Nonsymmetric Eigenvalue Problems II. Matrix Pairs," *Lin. Alg. and Its Applic. 197*, 283–295.

A. Ruhe (1994). "The Rational Krylov Algorithm for Nonsymmetric Eigenvalue Problems III: Complex Shifts for Real Matrices," *BIT 34*,165–176.

A. Ruhe and T. Wiberg (1972). "The Method of Conjugate Gradients Used in Inverse Iteration," *BIT 12*, 543–54.

H. Rutishauser (1958). "Solution of Eigenvalue Problems with the LR Transformation," *Nat. Bur. Stand. App. Math. Ser. 49*, 47–81.

H. Rutishauser (1966). "Bestimmung der Eigenwerte Orthogonaler Matrizen," *Numer. Math. 9*, 104–108.

H. Rutishauser (1966). "The Jacobi Method for Real Symmetric Matrices," *Numer. Math. 9*, 1–10. See also Wilkinson and Reinsch (1971, pp. 202–11).

H. Rutishauser (1970). "Simultaneous Iteration Method for Symmetric Matrices," *Numer. Math. 16*, 205–23. See also Wilkinson and Reinsch (1971,pp.284–302).

Y. Saad (1980). "On the Rates of Convergence of the Lanczos and the Block Lanczos Methods," *SIAM J. Num. Anal.17*, 687–706.

Y. Saad (1980). "Variations of Arnoldi's Method for Computing Eigenelements of Large Unsymmetric Matrices.," *Lin. Alg. and Its Applic. 34*, 269–295.

Y. Saad (1981). "Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems," *Math. Comp. 37*, 105–126.

Y. Saad (1982). "The Lanczos Biorthogonalization Algorithm and Other Oblique Projection Metods for Solving Large Unsymmetric Systems," *SIAM J. Numer. Anal. 19*, 485–506.

Y. Saad (1984). "Practical Use of Some Krylov Subspace Methods for Solving Indefinite and Nonsymmetric Linear Systems," *SIAM J. Sci. and Stat. Comp. 5*, 203–228.

Y. Saad (1985). "Practical Use of Polynomial Preconditionings for the Conjugate Gradient Method," *SIAM J. Sci. and Stat. Comp. 6*, 865–882.

Y. Saad (1986). "On the Condition Number of Some Gram Matrices Arising from Least Squares Approximation in the Complex Plane," *Numer. Math. 48*, 337–348.

Y. Saad (1987). "On the Lanczos Method for Solving Symmetric Systems with Several Right Hand Sides," *Math. Comp. 48*, 651–662.

Y. Saad (1988). "Preconditioning Techniques for Indefinite and Nonsymmetric Linear Systems," *J. Comput. Appl. Math. 24*, 89–105.

Y. Saad (1989). "Krylov Subspace Methods on Supercomputers," *SIAM J. Sci. and Stat. Comp. 10*, 1200–1322.

Y. Saad (1992). *Numerical Methods for Large Eigenvalue Problems: Theory and Algorithms*, John Wiley and Sons, New York.

Y. Saad (1993). "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM J. Sci. Comput. 14*, 461–469.

Y. Saad (1996). *Iterative Methods for Sparse Linear Systems*, PWS Publishing Co., Boston.

Y. Saad and M.H. Schultz (1985). "Conjugate Gradient-Like Algorithms for Solving Nonsymmetric Linear Systems," *Math. Comp. 44*, 417–424.

Y. Saad and M.H. Schultz (1986). "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Scientific and Stat. Comp. 7*, 856–869.

Y. Saad and M.H. Schultz (1989). "Data Communication in Parallel Architectures," *J. Dist. Parallel Comp. 11*, 131–150.

Y. Saad and M.H. Schultz (1989). "Data Communication in Hypercubes," *J. Dist. Parallel Comp. 6*, 115–135.

A. Sameh (1971). "On Jacobi and Jacobi-like Algorithms for a Parallel Computer," *Math. Comp. 25*, 579–90.

A. Sameh and D. Kuck (1978). "On Stable Parallel Linear System Solvers," *J. Assoc. Comp. Mach.* *25*, 81-91.

A. Sameh, J. Lermit and K. Noh (1975). "On the Intermediate Eigenvalues of Symmetric Sparse Matrices," *BIT 12*, 543–54.

M.A. Sanders (1995). "Solution of Sparse Rectangular Systems," *BIT 35*, 588–604.

M.A. Saunders, H.D. Simon, and E.L. Yip (1988). "Two Conjugate Gradient-Type Methods for Unsymmetric Linear Equations," *SIAM J. Num. Anal.* *25*, 927–940.

K. Schittkowski and J. Stoer (1979). "A Factorization Method for the Solution of Constrained Linear Least Squares Problems Allowing for Subsequent Data changes," *Numer. Math. 31*, 431–463.

W. Schönauer (1987). *Scientific Computing on Vector Computers*, North Holland, Amsterdam.

P. Schonemann (1966). "A Generalized Solution of the Orthogonal Procrustes Problem," *Psychometrika 31*, 1–10.

A. Schonhage (1964). "On the Quadratic Convergence of the Jacobi Process," *Numer. Math. 6*, 410–12.

A. Schonhage (1979). "Arbitrary Perturbations of Hermitian Matrices," *Lin. Alg. and Its Applic. 24*, 143–49.

R.S. Schreiber (1986). "Solving Eigenvalue and Singular Value Problems on an Undersized Systolic Array," *SIAM J. Sci. and Stat. Comp. 7*, 441–451.

R.S. Schreiber (1988). "Block Algorithms for Parallel Machines," in *Numerical Algorithms for Modern Parallel Computer Architectures*, M.H. Schultz (ed), IMA Volumes in Mathematics and Its Applications, Number 13, Springer-Verlag, Berlin, 197–207.

R.S. Schreiber and B.N. Parlett (1987). "Block Reflectors: Theory and Computation," *SIAM J. Numer. Anal. 25*, 189-205.

R.S. Schreiber and C. Van Loan (1989). "A Storage-Efficient WY Representation for Products of Householder Transformations," *SIAM J. Sci. and Stat. Comp. 10*, 52–57.

M.H. Schultz, ed. (1988). *Numerical Algorithms for Modern Parallel Computer Architectures*, IMA Volumes in Mathematics and Its Applications, Number 13, Springer-Verlag, Berlin.

I. Schur (1909). "On the Characteristic Roots of a Linear Substitution with an Application to the Theory of Integral Equations." *Math. Ann. 66*, 488-510 (German).

H.R. Schwartz (1968). "Tridiagonalization of a Symmetric Band Matrix," *Numer. Math. 12*, 231–41. See also Wilkinson and Reinsch (1971, 273–83).

H.R. Schwartz (1974). "The Method of Coordinate Relaxation for $(A - \lambda B)x = 0$," *Num. Math. 23*, 135–52.

D. Scott (1978). "Analysis of the Symmetric Lanczos Process," Electronic Research Laboratory Technical Report UCB/ERL M78/40, University of California, Berkeley.

D.S. Scott (1979). "Block Lanczos Software for Symmetric Eigenvalue Problems," Report ORNL/CSD-48, Oak Ridge National Laboratory, Union Carbide Corporation, Oak Ridge, Tennessee.

D.S. Scott (1979). "How to Make the Lanczos Algorithm Converge Slowly," *Math. Comp. 33*, 239–47.

D.S. Scott (1984). "Computing a Few Eigenvalues and Eigenvectors of a Symmetric Band Matrix," *SIAM J. Sci. and Stat. Comp. 5*, 658–666.

D.S. Scott (1985). "On the Accuracy of the Gershgorin Circle Theorem for Bounding the Spread of a Real Symmetric Matrix," *Lin. Alg. and Its Applic. 65*, 147–155

D.S. Scott, M.T. Heath, and R.C. Ward (1986). "Parallel Block Jacobi Eigenvalue Algorithms Using Systolic Arrays," *Lin. Alg. and Its Applic. 77*, 345–356.

M.K. Seager (1986). "Parallelizing Conjugate Gradient for the Cray X-MP," *Parallel Computing 3*, 35–47.

J.J. Seaton (1969). "Diagonalization of Complex Symmetric Matrices Using a Modified Jacobi Method," *Comp. J. 12*, 156–57.

S. Serbin (1980). "On Factoring a Class of Complex Symmetric Matrices Without Pivoting," *Math. Comp. 35*, 1231-1234.

S. Serbin and S. Blalock (1979). "An Algorithm for Computing the Matrix Cosine," *SIAM J. Sci. Stat. Comp. 1*, 198-204.

J.W. Sheldon (1955). "On the Numerical Solution of Elliptic Difference Equations," *Math. Tables Aids Comp. 9*, 101–12.

W. Shougen and Z. Shuqin (1991). "An Algorithm for $Ax = \lambda Bx$ with Symmetric and Positive Definite A and B," *SIAM J. Matrix Anal. Appl. 12*, 654-660.

G. Shroff (1991). "A Parallel Algorithm for the Eigenvalues and Eigenvectors of a General Complex Matrix," *Numer. Math. 58*, 779–806.

G. Shroff and C.H. Bischof (1992). "Adaptive Condition Estimation for Rank-One Updates of QR Factorizations," *SIAM J. Matrix Anal. Appl. 13*, 1264–1278.

G. Shroff and R. Schreiber (1989). "On the Convergence of the Cyclic Jacobi Method for Parallel Block Orderings," *SIAM J. Matrix Anal. Appl. 10*, 326–346.

H. Simon (1984). "Analysis of the Symmetric Lanczos Algorithm with Reorthogonalization Methods," *Lin. Alg. and Its Applic. 61*, 101–132.

B. Singer and S. Spilerman (1976). "The Representation of Social Processes by Markov Models," *Amer. J. Sociology 82*, 1–54.

R.D. Skeel (1979). "Scaling for numerical stability in Gaussian Elimination," *J. ACM 26*, 494–526.

R.D. Skeel (1980). "Iterative Refinement Implies Numerical Stability for Gaussian Elimination," *Math. Comp. 35*, 817–832.

R.D. Skeel (1981). "Effect of Equilibration on Residual Size for Partial Pivoting," *SIAM J. Num. Anal. 18*, 449–55.

G.L.G. Sleijpen and D.R. Fokkema (1993). "BICGSTAB(ℓ) for Linear Equations Involving Unsymmetric Matrices with Complex Spectrum," *Electronic Transactions on Numerical Analysis 1*, 11–32.

B.T. Smith, J.M. Boyle, Y. Ikebe, V.C. Klema, and C.B. Moler (1970). *Matrix Eigensystem Routines: EISPACK Guide, 2nd ed.*, Lecture Notes in Computer Science, Volume 6, Springer-Verlag, New York.

R.A. Smith (1967). "The Condition Numbers of the Matrix Eigenvalue Problem," *Numer. Math. 10* 232–40.

F. Smithies (1970). *Integral Equations*, Cambridge University Press, Cambridge.

P. Sonneveld (1989). "CGS, A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems," *SIAM J. Sci. and Stat. Comp. 10*, 36–52.

D.C. Sorensen (1992). "Implicit Application of Polynomial Filters in a k-Step Arnoldi Method," *SIAM J. Matrix Anal. Appl. 13*, 357–385.

D.C. Sorensen (1995). "Implicitly Restarted Arnoldi/Lanczos Methods for Large Scale Eigenvalue Calculations," in *Proceedings of the ICASE/LaRC Workshop on Parallel Numerical Algorithms, May 23–25, 1994*, D.E. Keyes, A. Sameh, and V. Venkatakrishnan (eds), Kluwer.

G.W. Stewart (1969). "Accelerating The Orthogonal Iteration for the Eigenvalues of a Hermitian Matrix," *Numer. Math. 13*, 362–76.

G.W. Stewart (1970). "Incorporating Original Shifts into the QR Algorithm for Symmetric Tridiagonal Matrices," *Comm. ACM 13*, 365–67.

G.W. Stewart (1971). "Error Bounds for Approximate Invariant Subspaces of Closed Linear Operators," *SIAM. J. Num. Anal. 8*, 796–808.

G.W. Stewart (1972). "On the Sensitivity of the Eigenvalue Problem $Ax = \lambda Bx$," *SIAM J. Num. Anal. 9*, 669–86.

G.W. Stewart (1973). "Error and Perturbation Bounds for Subspaces Associated with Certain Eigenvalue Problems," *SIAM Review 15*, 727–64.

G.W. Stewart (1973). *Introduction to Matrix Computations*, Academic Press, New York.

G.W. Stewart (1973). "Conjugate Direction Methods for Solving Systems of Linear Equations," *Numer. Math. 21*, 284–97.

G.W. Stewart (1974). "The Numerical Treatment of Large Eigenvalue Problems," *Proc. IFIP Congress 74*, North-Holland, pp. 666–72.

G.W. Stewart (1975). "The Convergence of the Method of Conjugate Gradients at Isolated Extreme Points in the Spectrum," *Numer. Math. 24*, 85–93.

G.W. Stewart (1975). "Gershgorin Theory for the Generalized Eigenvalue Problem $Ax = \lambda Bx$," *Math. Comp.* 29, 600–606.

G.W. Stewart (1975). "Methods of Simultaneous Iteration for Calculating Eigenvectors of Matrices," in *Topics in Numerical Analysis II*, ed. John J.H. Miller, Academic Press, New York, pp. 185–96.

G.W. Stewart (1976). "The Economical Storage of Plane Rotations," *Numer. Math.* 25, 137–38.

G.W. Stewart (1976). "Simultaneous Iteration for Computing Invariant Subspaces of Non-Hermitian Matrices," *Numer. Math.* 25, 123–36.

G.W. Stewart (1976). "Algorithm 406: HQR3 and EXCHNG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of a Real Upper Hessenberg Matrix," *ACM Trans. Math. Soft.* 2, 275–80.

G.W. Stewart (1976). "A Bibliographical Tour of the Large Sparse Generalized Eigenvalue Problem," in *Sparse Matrix Computations*, ed., J.R. Bunch and D.J. Rose, Academic Press, New York.

G.W. Stewart (1977). "Perturbation Bounds for the QR Factorization of a Matrix," *SIAM J. Num. Anal.* 14, 509–18.

G.W. Stewart (1977). "On the Perturbation of Pseudo-Inverses, Projections and Linear Least Squares Problems," *SIAM Review* 19, 634–662.

G.W. Stewart (1977). "Sensitivity Coefficients for the Effects of Errors in the Independent Variables in a Linear Regression," Technical Report TR-571, Department of Computer Science, University of Maryland, College Park, MD.

G.W. Stewart (1978). "Perturbation Theory for the Generalized Eigenvalue Problem'", in *Recent Advances in Numerical Analysis*, ed. C. de Boor and G.H. Golub, Academic Press, New York.

G.W. Stewart (1979). "A Note on the Perturbation of Singular Values," *Lin. Alg. and Its Applic.* 28, 213–16.

G.W. Stewart (1979). "Perturbation Bounds for the Definite Generalized Eigenvalue Problem," *Lin. Alg. and Its Applic.* 23, 69–86.

G.W. Stewart (1979). "The Effects of Rounding Error on an Algorithm for Downdating a Cholesky Factorization," *J. Inst. Math. Applic.* 23, 203–13.

G.W. Stewart (1980). "The Efficient Generation of Random Orthogonal Matrices with an Application to Condition Estimators," *SIAM J. Num. Anal.* 17, 403–9.

G.W. Stewart (1981). "On the Implicit Deflation of Nearly Singular Systems of Linear Equations," *SIAM J. Sci. and Stat. Comp.* 2, 136-140.

G.W. Stewart (1983). "A Method for Computing the Generalized Singular Value Decomposition," in *Matrix Pencils*, ed. B. Kågström and A. Ruhe, Springer-Verlag, New York, pp. 207–20.

G.W. Stewart (1984). "A Second Order Perturbation Expansion for Small Singular Values," *Lin. Alg. and Its Applic.* 56, 231–236.

G.W. Stewart (1984). "Rank Degeneracy," *SIAM J. Sci. and Stat. Comp.* 5, 403–413.

G.W. Stewart (1984). "On the Asymptotic Behavior of Scaled Singular Value and QR Decompositions," *Math. Comp.* 43, 483–490.

G.W. Stewart (1985). "A Jacobi-Like Algorithm for Computing the Schur Decomposition of a Nonhermitian Matrix," *SIAM J. Sci. and Stat. Comp.* 6, 853–862.

G.W. Stewart (1987). "Collinearity and Least Squares Regression," *Statistical Science* 2, 68–100.

G.W. Stewart (1989). "On Scaled Projections and Pseudoinverses," *Lin. Alg. and Its Applic.* 112, 189–193.

G.W. Stewart (1992). "An Updating Algorithm for Subspace Tracking," *IEEE Trans. Signal Proc.* 40, 1535–1541.

G.W. Stewart (1993). "Updating a Rank-Revealing ULV Decomposition," *SIAM J. Matrix Anal. Appl.* 14, 494–499.

G.W. Stewart (1993). "On the Perturbation of LU Cholesky, and QR Factorizations," *SIAM J. Matrix Anal. Appl.* 14, 1141–1145.

G.W. Stewart (1993). "On the Early History of the Singular Value Decomposition," *SIAM Review 35*, 551-566.

G.W. Stewart (1994). "Perturbation Theory for Rectangular Matrix Pencils," *Lin. Alg. and Applic. 208/209*, 297-301.

G.W. Stewart (1994). "Updating URV Decompositions in Parallel," *Parallel Computing 20*, 151-172.

G.W. Stewart and J.-G. Sun (1990). *Matrix Perturbation Theory*, Academic Press, San Diego.

G.W. Stewart and G. Zheng (1991). "Eigenvalues of Graded Matrices and the Condition Numbers of Multiple Eigenvalues," *Numer. Math. 58*, 703-712.

M. Stewart and P. Van Dooren (1996). "Stability Issues in the Factorization of Structured Matrices," *SIAM J. Matrix Anal. Appl. 18*, to appear.

H.S. Stone (1973). "An Efficient Parallel Algorithm for the Solution of a Tridiagonal Linear System of Equations," *J. ACM 20*, 27-38.

H.S. Stone (1975). "Parallel Tridiagonal Equation Solvers," *ACM Trans. Math. Soft. 1*, 289-307.

G. Strang (1988). "A Framework for Equilibrium Equations," *SIAM Review 30*, 283-297.

G. Strang (1993). *Introduction to Linear Algebra*, Wellesley-Cambridge Press, Wellesley MA.

V. Strassen (1969). "Gaussian Elimination is Not Optimal," *Numer. Math. 13*, 354-356.

J.-G. Sun (1982). "A Note on Stewart's Theorem for Definite Matrix Pairs," *Lin. Alg. and Its Applic. 48*, 331-339.

J.-G. Sun (1983). "Perturbation Analysis for the Generalized Singular Value Problem," *SIAM J. Numer. Anal. 20*, 611-625.

J.-G. Sun (1992). "On Condition Numbers of a Nondefective Multiple Eigenvalue," *Numer. Math. 61*, 265-276.

J.-G. Sun (1992). "Rounding Error and Perturbation Bounds for the Cholesky and LDL^T Factorizations," *Lin. Alg. and Its Applic. 173*, 77-97.

J.-G. Sun (1995). "A Note on Backward Error Perturbations for the Hermitian Eigenvalue Problem," *BIT 35*, 385-393.

J.-G. Sun (1995). "On Perturbation Bounds for the QR Factorization," *Lin. Alg. and Its Applic. 215*, 95-112.

X. Sun and C.H. Bischof (1995). "A Basis-Kernel Representation of Orthogonal Matrices," *SIAM J. Matrix Anal. Appl. 16*, 1184-1196.

P.N. Swarztrauber (1979). "A Parallel Algorithm for Solving General Tridiagonal Equations," *Math. Comp. 33*, 185-199.

P.N. Swarztrauber and R.A. Sweet (1973). "The Direct Solution of the Discrete Poisson Equation on a Disk," *SIAM J. Num. Anal. 10*, 900-907.

P.N. Swarztrauber and R.A. Sweet (1989). "Vector and Parallel Methods for the Direct Solution of Poisson's Equation," *J. Comp. Appl. Math. 27*, 241-263.

D.R. Sweet (1991). "Fast Block Toeplitz Orthogonalization," *Numer. Math. 58*, 613-629.

D.R. Sweet (1993). "The Use of Pivoting to Improve the Numerical Performance of Algorithms for Toeplitz Matrices," *SIAM J. Matrix Anal. Appl. 14*, 468-493.

R.A. Sweet (1974). "A Generalized Cyclic Reduction Algorithm," *SIAM J. Num. Anal. 11*, 506-20.

R.A. Sweet (1977). "A Cyclic Reduction Algorithm for Solving Block Tridiagonal Systems of Arbitrary Dimension," *SIAM J. Num. Anal. 14*, 706-20.

H.J. Symm and J.H. Wilkinson (1980). "Realistic Error Bounds for a Simple Eigenvalue and Its Associated Eigenvector," *Numer. Math. 35*, 113-26.

P.T.P. Tang (1994). "Dynamic Condition Estimation and Rayleigh-Ritz Approximation," *SIAM J. Matrix Anal. Appl. 15*, 331-346.

R.A. Tapia and D.L. Whitley (1988). "The Projected Newton Method Has Order $1+\sqrt{2}$ for the Symmetric Eigenvalue Problem," *SIAM J. Num. Anal. 25*, 1376-1382.

G.L. Thompson and R.L. Weil (1970). "Reducing the Rank of $A - \lambda B$," *Proc. Amer. Math. Sec. 26*, 548-54.

G.L. Thompson and R.L. Weil (1972). "Roots of Matrix Pencils $Ay = \lambda By$: Existence, Calculations, and Relations to Game Theory," *Lin. Alg. and Its Applic. 5*, 207–26.

M.J. Todd (1990). "A Dantzig-Wolfe-like Variant of Karmarker's Interior-Point Linear Programming Algorithm," *Operations Research 38*, 1006–1018.

K.-C. Toh and L.N. Trefethen (1994). "Pseudozeros of Polynomials and Pseudospectra of Companion Matrices," *Numer. Math. 68*, 403–425.

L.N. Trefethen (1992). "Pseudospecta of Matrices," in *Numerical Analysis 1991*, D.F. Griffiths and G.A. Watson (eds), Longman Scientific and Technical, Harlow, Essex, UK, 234–262.

L.N. Trefethen and D. Bau III (1997). *Numerical Linear Algebra*, SIAM Publications, Philadelphia, PA.

L.N. Trefethen and R.S. Schreiber (1990). "Average-Case Stability of Gaussian Elimination," *SIAM J. Matrix Anal. Appl. 11*, 335–360.

L.N. Trefethen, A.E. Trefethen, S.C. Reddy, and T.A. Driscoll (1993). "Hydrodynamic Stability Without Eigenvalues," *Science 261*, 578–584.

W.F. Trench (1964). "An Algorithm for the Inversion of Finite Toeplitz Matrices," *J. SIAM 12*, 515–22.

W.F. Trench (1989). "Numerical Solution of the Eigenvalue Problem for Hermitian Toeplitz Matrices," *SIAM J. Matrix Anal. Appl. 10*, 135–146.

N.K. Tsao (1975). "A Note on Implementing the Householder Transformations." *SIAM J. Num. Anal. 12*, 53–58.

H.W. Turnbull and A.C. Aitken (1961). *An Introduction to the Theory of Canonical Matrices*, Dover Publications, New York, pp. 102–5.

F. Uhlig (1973). "Simultaneous Block Diagonalization of Two Real Symmetric Matrices," *Lin. Alg. and Its Applic. 7*, 281–89.

F. Uhlig (1976). "A Canonical Form for a Pair of Real Symmetric Matrices That Generate a Nonsingular Pencil," *Lin. Alg. and Its Applic. 14*, 189–210.

R. Underwood (1975). "An Iterative Block Lanczos Method for the Solution of Large Sparse Symmetric Eigenproblems," Report STAN-CS-75-495, Department of Computer Science, Stanford University, Stanford, California.

R.J. Vaccaro, ed. (1991). *SVD and Signal Processing II: Algorithms, Analysis, and Applications*. Elsevier, Amsterdam.

R.J. Vaccaro (1994). "A Second-Order Perturbation Expansion for the SVD," *SIAM J. Matrix Anal. Applic. 15*, 661–671.

R.A. Van De Geijn (1993). "Deferred Shifting Schemes for Parallel QR Methods," *SIAM J. Matrix Anal. Appl. 14*, 180–194.

J. Vandergraft (1971). "Generalized Rayleigh Methods with Applications to Finding Eigenvalues of Large Matrices," *Lin. Alg. and Its Applic. 4*, 353–68.

A. Van der Sluis (1969). "Condition Numbers and Equilibration Matrices," *Numer. Math. 14*, 14–23.

A. Van der Sluis (1970). "Condition, Equilibration, and Pivoting in Linear Algebraic Systems," *Numer. Math. 15*, 74–86.

A. Van der Sluis (1975). "Stability of the Solutions of Linear Least Squares Problem," *Numer. Math. 23*, 241–54.

A. Van der Sluis (1975). "Perturbations of Eigenvalues of Non-normal Matrices," *Comm. ACM 18*, 30–36.

A. Van der Sluis and H.A. Van Der Vorst (1986). "The Rate of Convergence of Conjugate Gradients," *Numer. Math. 48*, 543–560.

A. Van der Sluis and G.W. Veltkamp (1979). "Restoring Rank and Consistency by Orthogonal Projection," *Lin. Alg. and Its Applic. 28*, 257–78.

H. Van de Vel (1977). "Numerical Treatment of a Generalized Vandermonde systems of Equations," *Lin. Alg. and Its Applic. 17*, 149–74.

E.F. Van de Velde (1994). *Concurrent Scientific Computing*, Springer-Verlag, New York.

H.A. Van der Vorst (1982). "A Vectorizable Variant of Some ICCG Methods," *SIAM J. Sci. and Stat. Comp. 3*, 350–356.

H.A. Van der Vorst (1982). "A Generalized Lanczos Scheme," *Math. Comp. 39*, 559–562.

H.A. Van der Vorst (1986). "The Performance of Fortran Implementations for Preconditioned Conjugate Gradients on Vector Computers," *Parallel Computing 3*, 49–58.

H.A. Van der Vorst (1986). "An Iterative Solution Method for Solving $f(A)x = b$ Using Krylov Subspace Information Obtained for the Symmetric Positive Definite Matrix A," *J. Comp. and App. Math. 18*, 249–263.

H. Van der Vorst (1987). "Large Tridiagonal and Block Tridiagonal Linear Systems on Vector and Parallel Computers," *Parallel Comput. 5*, 45–54.

H. Van Der Vorst (1989). "High Performance Preconditioning," *SIAM J. Sci. and Stat. Comp. 10*, 1174–1185.

H.A. Van Der Vorst (1992). "BiCGSTAB: A Fast and Smoothly Converging Variant of the Bi-CG for the Solution of Nonsymmetric Linear Systems," *SIAM J. Sci. and Stat. Comp. 13*, 631–644.

P. Van Dooren (1979). "The Computation of Kronecker's Canonical Form of a Singular Pencil," *Lin. Alg. and Its Applic. 27*, 103–40.

P. Van Dooren (1981). "A Generalized Eigenvalue Approach for Solving Riccati Equations," *SIAM J. Sci. and Stat. Comp. 2*, 121–135.

P. Van Dooren (1981). "The Generalized Eigenstructure Problem in Linear System Theory," *IEEE Trans. Auto. Cont. AC-26*, 111–128.

P. Van Dooren (1982). "Algorithm 590: DSUBSP and EXCHQZ: Fortran Routines for Computing Deflating Subspaces with Specified Spectrum," *ACM Trans. Math. Software 8*, 376–382.

S. Van Huffel (1992). "On the Significance of Nongeneric Total Least Squares Problems," *SIAM J. Matrix Anal. Appl. 13*, 20–35.

S. Van Huffel and H. Park (1994). "Parallel Tri- and Bidiagonalization of Bordered Bidiagonal Matrices," *Parallel Computing 20*, 1107–1128.

S. Van Huffel and J. Vandewalle (1987). "Subset Selection Using the Total Least Squares Approach in Collinearity Problems with Errors in the Variables," *Lin. Alg. and Its Applic. 88/89*, 695–714.

S. Van Huffel and J. Vandewalle (1988). "The Partial Total Least Squares Algorithm," *J. Comp. and App. Math. 21*, 333–342.

S. Van Huffel and J. Vandewalle (1988). "Analysis and Solution of the Nongeneric Total Least Squares Problem," *SIAM J. Matrix Anal. Appl. 9*, 360–372.

S. Van Huffel and J. Vandewalle (1989). "Analysis and Properties of the Generalized Total Least Squares Problem $AX \approx B$ When Some or All Columns in A are Subject to Error," *SIAM J. Matrix Anal. Appl. 10*, 294–315.

S. Van Huffel and J. Vandewalle (1991). *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM Publications, Philadelphia, PA.

S. Van Huffel, J. Vandewalle, and A. Haegemans (1987). "An Efficient and Reliable Algorithm for Computing the Singular Subspace of a Matrix Associated with its Smallest Singular Values," *J. Comp. and Appl. Math. 19*, 313–330.

S. Van Huffel and H. Zha (1991). "The Restricted Total Least Squares Problem: Formulation, Algorithm, and Properties," *SIAM J. Matrix Anal. Appl. 12*, 292–309.

S. Van Huffel and H. Zha (1993). "An Efficient Total Least Squares Algorithm Based On a Rank-Revealing Two-Sided Orthogonal Decomposition," *Numerical Algorithms 4*, 101–133.

H.P.M. van Kempen (1966). "On Quadratic Convergence of the Special Cyclic Jacobi Method," *Numer. Math. 9*, 19–22.

C.F. Van Loan (1973). "Generalized Singular Values With Algorithms and Applications," Ph.D. thesis, University of Michigan, Ann Arbor.

C.F. Van Loan (1975). "A General Matrix Eigenvalue Algorithm," *SIAM J. Num. Anal. 12*, 819–834.

C.F. Van Loan (1975). "A Study of the Matrix Exponential," Numerical Analysis Report No. 10, Dept. of Maths., University of Manchester, England.

C.F. Van Loan (1976). "Generalizing the Singular Value Decomposition," *SIAM J. Num. Anal. 13,* 76–83.

C.F. Van Loan (1977). "On the Limitation and Application of Padé Approximation to the Matrix Exponential," in *Padé and Rational Approximation*, ed. E.B. Saff and R.S. Varga, Academic Press, New York.

C.F. Van Loan (1977). "The Sensitivity of the Matrix Exponential," *SIAM J. Num. Anal. 14,* 971–81.

C.F. Van Loan (1978). "Computing Integrals Involving the Matrix Exponential," *IEEE Trans. Auto. Cont. AC-23,* 395–404.

C.F. Van Loan (1978). "A Note on the Evaluation of Matrix Polynomials," *IEEE Trans. Auto. Cont. AC-24,* 320–21.

C.F. Van Loan (1982). "Using the Hessenberg Decomposition in Control Theory," in *Algorithms and Theory in Filtering and Control*, D.C. Sorensen and R.J. Wets (eds), Mathematical Programming Study No. 18, North Holland, Amsterdam, pp. 102–11.

C.F. Van Loan (1984). "A Symplectic Method for Approximating All the Eigenvalues of a Hamiltonian Matrix," *Lin. Alg. and Its Applic. 61,* 233–252.

C.F. Van Loan (1985). "How Near is a Stable Matrix to an Unstable Matrix?," *Contemporary Mathematics, Vol. 47,* 465–477.

C.F. Van Loan (1985). "On the Method of Weighting for Equality Constrained Least Squares Problems," *SIAM J. Numer. Anal. 22,* 851–864.

C.F. Van Loan (1985). "Computing the CS and Generalized Singular Value Decomposition," *Numer. Math. 46,* 479–492.

C.F. Van Loan (1987). "On Estimating the Condition of Eigenvalues and Eigenvectors," *Lin. Alg. and Its Applic.* 88/89, 715–732.

C.F. Van Loan (1992). *Computational Frameworks for the Fast Fourier Transform*, SIAM Publications, Philadelphia, PA.

C.F. Van Loan (1997). *Introduction to Scientific Computing: A Matrix-Vector Approach Using Matlab*, Prentice Hall, Upper Saddle River, NJ.

J.M. Varah (1968). "The Calculation of the Eigenvectors of a General Complex Matrix by Inverse Iteration," *Math. Comp. 22,* 785–91.

J.M. Varah (1968). "Rigorous Machine Bounds for the Eigensystem of a General Complex Matrix," *Math. Comp. 22,* 793–801.

J.M. Varah (1970). "Computing Invariant Subspaces of a General Matrix When the Eigensystem is Poorly Determined," *Math. Comp. 24,* 137–49.

J.M. Varah (1972). "On the Solution of Block-Tridiagonal Systems Arising from Certain Finite-Difference Equations," *Math. Comp. 26,* 859–68.

J.M. Varah (1973). "On the Numerical Solution of Ill-Conditioned Linear Systems with Applications to Ill-Posed Problems," *SIAM J. Num. Anal. 10,* 257–67.

J.M. Varah (1979). "On the Separation of Two Matrices," *SIAM J. Num. Anal. 16,* 212–22.

J.M. Varah (1993). "Errors and Perturbations in Vandermonde Systems," *IMA J. Num. Anal. 13,* 1–12.

J.M. Varah (1994). "Backward Error Estimates for Toeplitz Systems," *SIAM J. Matrix Anal. Appl. 15,* 408–417.

R.S. Varga (1961). "On Higher-Order Stable Implicit Methods for Solving Parabolic Partial Differential Equations," *J. Math. Phys. 40,* 220–31.

R.S. Varga (1962). *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ.

R.S. Varga (1970). "Minimal Gershgorin Sets for Partitioned Matrices," *SIAM J. Num. Anal. 7,* 493–507.

R.S. Varga (1976). "On Diagonal Dominance Arguments for Bounding $\| A^{-1} \|_\infty$," *Lin. Alg. and Its Applic. 14,* 211–17.

S.A. Vavasis (1994). "Stable Numerical Algorithms for Equilibrium Systems," *SIAM J. Matrix Anal. Appl. 15,* 1108–1131.

S.A. Vavasis (1992). "Preconditioning for Boundary Integral Equations," *SIAM J. Matrix Anal. Appl. 13,* 905–925.

K. Veselić (1993). "A Jacobi Eigenreduction Algorithm for Definite Matrix Pairs," *Numer. Math. 64*, 241–268.

K. Veselić and V. Hari (1989). "A Note on a One-Sided Jacobi Algorithm," *Numer. Math. 56*, 627–633.

W.J. Vetter (1975). "Vector Structures and Solutions of Linear Matrix Equations," *Lin. Alg. and Its Applic. 10*, 181–88.

C. Vuik and H.A. van der Vorst (1992). "A Comparison of Some GMRES-like Methods," *Lin. Alg. and Its Applic. 160*, 131–162.

A. Wald (1940). "The Fitting of Straight Lines if Both Variables are Subject to Error," *Annals of Mathematical Statistics 11*, 284–300.

B. Waldén, R. Karlson, J. Sun (1995). "Optimal Backward Perturbation Bounds for the Linear Least Squares Problem," *Numerical Lin. Alg. with Applic. 2*, 271–286.

H.F. Walker (1988). "Implementation of the GMRES Method Using Householder Transformations," *SIAM J. Sci. Stat. Comp. 9*, 152-163.

R.C. Ward (1975). "The Combination Shift QZ Algorithm," *SIAM J. Num. Anal. 12*, 835–853.

R.C. Ward (1977). "Numerical Computation of the Matrix Exponential with Accuracy Estimate," *SIAM J. Num. Anal. 14*, 600–14.

R.C. Ward (1981). "Balancing the Generalized Eigenvalue Problem," *SIAM J. Sci and Stat. Comp. 2*, 141–152.

R.C. Ward and L.J. Gray (1978). "Eigensystem Computation for Skew-Symmetric and A Class of Symmetric Matrices," *ACM Trans. Math. Soft. 4*, 278–85.

D.S. Watkins (1982). "Understanding the QR Algorithm," *SIAM Review 24*, 427–440.

D.S. Watkins (1991). *Fundamentals of Matrix Computations*, John Wiley and Sons, New York.

D.S. Watkins (1993). "Some Perspectives on the Eigenvalue Problem," *SIAM Review 35*, 430–471.

D.S. Watkins and L. Elsner (1991). "Chasing Algorithms for the Eigenvalue Problem," *SIAM J. Matrix Anal. Appl. 12*, 374–384.

D.S. Watkins and L. Elsner (1991). "Convergence of Algorithms of Decomposition Type for the Eigenvalue Problem," *Lin.Alg. and Its Applic. 143*, 19–47.

D.S. Watkins and L. Elsner (1994). "Theory of Decomposition and Bulge-Chasing Algorithms for the Generalized Eigenvalue Problem," *SIAM J. Matrix Anal. Appl. 15*, 943–967.

G.A. Watson (1988). "The Smallest Perturbation of a Submatrix which Lowers the Rank of the Matrix," *IMA J. Numer. Anal. 8*, 295–304.

P.A. Wedin (1972). "Perturbation Bounds in Connection with the Singular Value Decomposition," *BIT 12*, 99–111.

P.A. Wedin (1973). "Perturbation Theory for Pseudo-Inverses," *BIT 13*, 217–32.

P.A. Wedin (1973). "On the Almost Rank-Deficient Case of the Least Squares Problem," *BIT 13*, 344–54.

M. Wei (1992). "Perturbation Theory for the Rank-Deficient Equality Constrained Least Squares Problem," *SIAM J. Num. Anal. 29*, 1462–1481.

M. Wei (1992). "Algebraic Properties of the Rank-Deficient Equality-Constrained and Weighted Least Squares Problems," *Lin. Alg. and Its Applic. 161*, 27–44.

M. Wei (1992). "The Analysis for the Total Least Squares Problem with More than One Solution," *SIAM J. Matrix Anal. Appl. 13*, 746–763.

O. Widlund (1978). "A Lanczos Method for a Class of Nonsymmetric Systems of Linear Equations," *SIAM J. Numer. Anal. 15*, 801–12.

J.H. Wilkinson (1961). "Error Analysis of Direct Methods of Matrix Inversion," *J. ACM 8*, 281-330.

J.H. Wilkinson (1962). "Note on the Quadratic Convergence of the Cyclic Jacobi Process," *Numer. Math. 6*, 296–300.

J.H. Wilkinson (1963). *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, NJ.

J.H. Wilkinson (1965). *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, England.

J.H. Wilkinson (1965). "Convergence of the LR, QR, and Related Algorithms," *Comp. J. 8*, 77–84.

J.H. Wilkinson (1968). "Global Convergence of Tridiagonal QR Algorithm With Origin Shifts," *Lin. Alg. and Its Applic. 1*, 409–20.

J.H. Wilkinson (1968). "Almost Diagonal Matrices with Multiple or Close Eigenvalues," *Lin. Alg. and Its Applic. 1*, 1–12.

J.H. Wilkinson (1968). "À Priori Error Analysis of Algebraic Processes," *Proc. International Congress Math.* (Moscow: Izdat. Mir, 1968), pp. 629–39.

J.H. Wilkinson (1971). "Modern Error Analysis," *SIAM Review 13*, 548–68.

J.H. Wilkinson (1972). "Note on Matrices with a Very Ill-Conditioned Eigenproblem," *Numer. Math. 19*, 176–78.

J.H. Wilkinson (1977). "Some Recent Advances in Numerical Linear Algebra," in *The State of the Art in Numerical Analysis*, ed. D.A.H. Jacobs, Academic Press, New York, pp. 1–53.

J.H. Wilkinson (1978). "Linear Differential Equations and Kronecker's Canonical Form," in *Recent Advances in Numerical Analysis* , ed. C. de Boor and G.H. Golub, Academic Press, New York, pp. 231–65.

J.H. Wilkinson (1979). "Kronecker's Canonical Form and the QZ Algorithm," *Lin. Alg. and Its Applic. 28*, 285–303.

J.H. Wilkinson (1984). "On Neighboring Matrices with Quadratic Elementary Divisors," *Numer. Math. 44*, 1–21.

J.H. Wilkinson and C. Reinsch, eds. (1971). *Handbook for Automatic Computation, Vol. 2, Linear Algebra*, Springer-Verlag, New York.

H. Wimmer and A.D. Ziebur (1972). "Solving the Matrix Equations $\Sigma f_p(A)g_p(A) = C$," *SIAM Review 14*, 318–23.

S. Winograd (1968). "A New Algorithm for Inner Product," *IEEE Trans. Comp. C-17*, 693–694.

M. Wolfe (1996). *High Performance Compilers for Parallel Computers*, Addison-Wesley, Reading MA.

A. Wouk, ed. (1986). *New Computing Environments: Parallel, Vector, and Systolic, SIAM Publications*, Philadelphia, PA.

H. Wozniakowski (1978). "Roundoff-Error Analysis of Iterations for Large Linear Systems," *Numer. Math. 30*, 301–314.

H. Wozniakowski (1980). "Roundoff Error Analysis of a New Class of Conjugate Gradient Algorithms," *Lin. Alg. and Its Applic. 29*, 507–29.

A. Wragg (1973). "Computation of the Exponential of a Matrix I: Theoretical Considerations," *J. Inst. Math. Applic. 11*, 369–75.

A. Wragg (1975). "Computation of the Exponential of a Matrix II: Practical Considerations," *J. Inst. Math. Applic. 15*, 273–78.

S.J. Wright (1993). "A Collection of Problems for Which Gaussian Elimination with Partial Pivoting is Unstable," *SIAM J. Sci. and Stat. Comp. 14*, 231–238.

J.M. Yohe (1979). "Software for Interval Arithmetic: A Reasonable Portable Package," *ACM Trans. Math. Soft. 5*, 50–63.

D.M. Young (1970). "Convergence Properties of the Symmetric and Unsymmetric Over-Relaxation Methods," *Math. Comp. 24*, 793–807.

D.M. Young (1971). *Iterative Solution of Large Linear Systems*, Academic Press, New York.

D.M. Young (1972). "Generalization of Property A and Consistent Ordering," *SIAM J. Num. Anal. 9*, 454–63.

D.M. Young and K.C. Jea (1980). "Generalized Conjugate Gradient Acceleration of Nonsymmetrizable Iterative Methods," *Lin. Alg. and Its Applic. 34*, 159–94.

L. Yu. Kolotilina and A. Yu. Yeremin (1993). "Factorized Sparse Approximate Inverse Preconditioning I: Theory," *SIAM J. Matrix Anal. Applic. 14*, 45–58.

L. Yu. Kolotilina and A. Yu. Yeremin (1995). "Factorized Sparse Approximate Inverse Preconditioning II: Solution of 3D FE Systems on Massively Parallel Computers," *Intern. J. High Speed Comput.* *7*, 191–215.

H. Zha (1991). "The Restricted Singular Value Decomposition of Matrix Triplets," *SIAM J. Matrix Anal. Appl.* *12*, 172–194.

H. Zha (1992). "A Numerical Algorithm for Computing the Restricted Singular Value Decomposition of Matrix Triplets," *Lin.Alg. and Its Applic.* *168*, 1–25.

H. Zha (1993). "A Componentwise Perturbation Analysis of the QR Decomposition," *SIAM J. Matrix Anal. Appl.* *4*, 1124–1131.

H. Zha and Z. Zhang (1995). "A Note on Constructing a Symmetric Matrix with Specified Diagonal Entries and Eigenvalues," *BIT* *35*, 448–451.

H. Zhang and W.F. Moss (1994). "Using Parallel Banded Linear System Solvers in Generalized Eigenvalue Problems," *Parallel Computing* *20*, 1089–1106.

Y. Zhang (1993). "A Primal-Dual Interior Point Approach for Computing the L_1 and L_∞ Solutions of Overdetermined Linear Systems," *J. Optimization Theory and Applications* *77*, 323–341.

S. Zohar (1969). "Toeplitz Matrix Inversion: The Algorithm of W.F. Trench," *J. ACM* *16*, 592–601.

Index